

DEVELOPMENT OF A BEAM LOSS MEASUREMENT SYSTEM WITH GIGABIT ETHERNET READOUT AT CERN

M. Kwiatkowski, M. Alsdorf, E. Angelogiannopoulos, B. Dehning, S. Jackson, W. Vigano, C. Zamantzas, CERN, Geneva, Switzerland

Abstract

The aim of the BLM Dual Polarity module under development at the European Organisation for Nuclear Research (CERN) is to measure and digitise with high precision the current produced by several types of beam loss detectors. In its default configuration, it is expected to provide data to the processing electronics through two point-to-point connections with bidirectional multi-gigabit optical links. For the development phases as well as later serving as a stand-alone measurement system, its reconfigurable FPGA device is exploited to provide a soft-core CPU with a custom made server. This server, running on the CPU, will expose through the Gigabit Ethernet connection and the TCP/IP protocol different types of data in the network. In this paper the development of the system and of the communication protocol is explored as well as the accompanying client application that is realised with the purpose of commanding, collecting storing and viewing the different types of data.

INTRODUCTION

During the upgrade of each injector line at the European Organisation for Nuclear Research (CERN), an up-to-date Beam Loss Monitoring (BLM) system will be included for the monitoring of the beam losses and machine protection. For this reason, a new BLM system is under design [1].

The BLM Dual Polarity (BLEDP) module is the first stage of that system with the purpose to collect and digitise the current output from the detector. The acquisition crate will be able to host up to eight of these modules. The BLEDP module should be able to digitise input current from eight input channels each having a wide range from 10 pA up to 200 mA.

To accomplish this, the range is split into two partially overlapping sub-ranges and for each of them a different circuit and measurement method is used. The current from 100 μ A up to 200 mA will be measured directly by the ADC as a voltage drop on the input resistor. It is so called Direct Analogue to Digital Conversion (DADC) method. The current in the lower range from 10 pA to 10 mA will be measured by making use of a low noise Fully Differential Frequency Converter (FDFC). More about the analogue front-ends used in both of the measurement methods can be found in [2].

The BLEDP module is equipped with a Cyclone 4GX150 FPGA device which is responsible for processing the FDFC data [3] and controlling the analogue circuitries. In the standalone version of the module, an Ethernet server is implemented in addition in the FPGA. To achieve this,

the Nios-II soft-core processor which is running under control of a μ C/OS-II real-time operating system is realised in the FPGA device. In this configuration, each of the modules in the acquisition crate have a separate link to the network. The measurement data collected is send to a dedicated supervision application which is used for online view as well as offline data storage and analysis. This paper focuses on the Ethernet implementation in the FPGA and it describes briefly the client application developed.

DATA TRANSMISSION

The data frame, shown in Fig. 1, was designed to satisfy both of the measurement methods which are automatically selected by the BLEDP module. That selection of the measurement method depends on the input current. In addition, the protocol must be capable of transmitting acquisition data from a single or multiple channels. The channel number or multi-channel transmission is requested by a command send by connected client. The measurement method and the channel information are attached as a header to the data sample which is acquired every 2 μ s and transmitted to the client. That is, each BLEDP frame will be 32 bits wide in total and it will consist of the header and the measurement data. The data rate per channel is therefore around 16 Mbit/s. The size of the data depend on the measurement method. The FDFC mode requires a 20 bit data field, whereas the DADC mode requires only a 16 bit. At the time of the writing, the DADC mode as well as the automated measurement method switching is not yet fully supported by the BLEDP firmware. Therefore, the size of the data field in this mode could be changed in the final specification. The data frames of the BLEDP are encapsulated in TCP/IP frames. The TCP protocol was selected due to its high reliability and automatic data reordering at the receiver side. However, it is expected to be also necessary to transmit data frames encapsulated into UDP format for the case of multichannel and raw data transmissions.

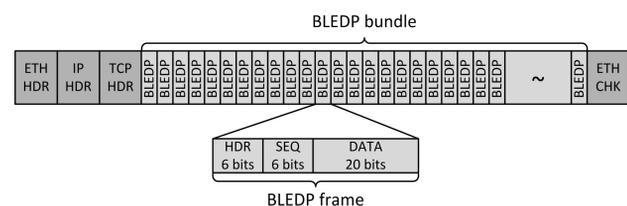


Figure 1: Structure of the BLM Data Frame transmitted over an Ethernet packet.

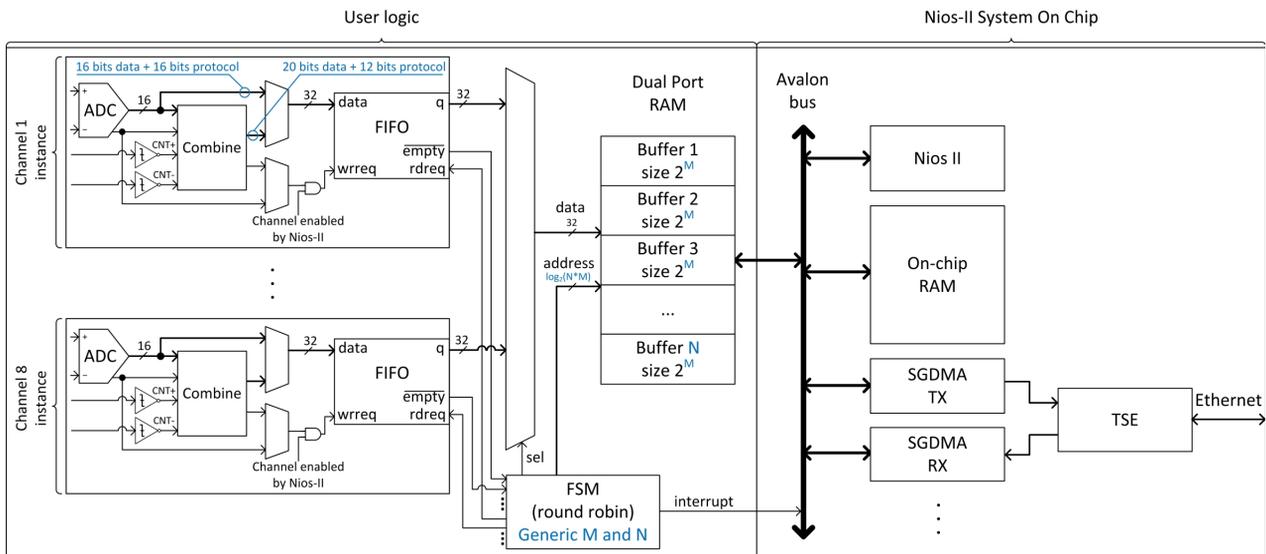


Figure 2: Architecture of the Nios-II system used for the Gigabit Ethernet Readout in the BLEDP firmware.

The remaining bits in the header part of the frame are used to include an incremental sequence number. In the frame there are 6 bits for the sequence number. That is, it will increment from 0 to 63 and will roll over as soon as it reaches the maximum value. This number is used by the client application to verify data continuity, which could potentially be broken due to not sufficient throughput of the network. The server implemented for the Nios-II soft processor is able to detect the low network throughput and it will try to recover from such situation.

The data frames are not transmitted one at a time but they are collected first in a buffer, whose size was selected so that its contents fit into a TCP Maximum Segment Size (MSS) packet, and the complete buffer is send every time it becomes full. That was necessary in order to avoid IP fragmentation, which would lower the performance of the Nios-II based TCP/IP software stack. The buffer size is set to 1400 bytes, i.e. contains 350 BLEDP frames. That package of the frames is referred to as “BLEDP bundle”.

Server Commands

A set of commands has been defined to be send to the BLEDP module by the client application for controlling and initiating the data transmission. Each command consists of 4 bytes divided into bit fields which specify:

- *Start or Stop transmission request.* A stop request disables all active transfers regardless of the other bit fields.
- *Data type selection.* It is possible to request data from one selected channel or from all the channels. In special mode, the raw ADC data from just one channel can be requested.
- *Channel number selection.* It is only valid when single channel data types are requested.
- *Data amount request.* It specifies the number of data

bundles which should be transmitted by the server. By taking into account fixed sampling time of the BLEDP module, i.e. $2 \mu s$, the data size can be easily transformed into an acquisition time. The maximum acquisition time that can be requested is approximately 13 hours.

On the reception of the command the server starts transmission of the selected channel and will end the transmission as soon as the requested amount of data has been transmitted.

Data Frames

The BLEDP data frames are equipped with a header that identifies the measurement method used to digitise the data as well as the channel source of the measurement data. It takes into account future development of the BLEDP module which will transmit also data collected from the several sensors included in the board to provide a status overview of the acquisition module. Therefore, the header is divided into three fields providing the following information:

- *Data type* provided, i.e. measurements or statuses.
- *Measurement method* specifying if the FDFC or DADC circuit was used to measure the input current.
- *Acquisition channel* specifying the source of the measurement data.

FPGA FIRMWARE

The FPGA firmware realised for the BLEDP module is comprised by two distinct parts. One part is the custom user logic and the second is the System-On-Chip (SOC) components generated with the assistance of the Altera development tools. A block diagram of the BLEDP firmware is shown in Fig. 2.



Figure 3: Example of the online data viewer in the client application.

User Logic

The user logic mainly consists of an instance for the data combination logic for each input channel. Each instance includes a FIFO queue which carries either FDFC or DADC data depending on the input current of the given channel. The data to be written in the FIFO is selected using a multiplexer. The DADC data path uses less bits and therefore there is more bits for the protocol. These bits are used to increase the range of the sequence number.

A Finite State Machine (FSM), implemented in the user logic, prepares the buffer for the TCP or UDP transmission. The FSM realises the round robin algorithm to read and write the data from the active channels into the buffer. The data buffers correspond to the BLEDP bundles and they are stored in a Dual Port RAM (DPRAM) memory. The FSM is configurable by generic parameters specifying the bundle size (M) and the number of the buffers (N). A multiplexer is used to select the channel to be written into the memory via one of the ports. The second port of the DPRAM is connected to the SOC via an Avalon bus. In the case of a multichannel transmission, the BLEDP packets are mixed in the BLEDP bundle. As shown previously, the packet header will allow the receiver later to distinguish their source. Whenever the buffer becomes full, the interrupt for the Nios-II is generated. The custom server, as soon as it receives the interrupt, is transmitting the contents of the buffer. It is also responsible for detecting faulty situations, e.g. data overwritten by the custom logic before being transmitted to the client.

System-On-Chip

The Nios-II SOC was created using the Altera Corp. standard IP cores and development tools. The processor is using only internal on-chip RAM memory for the program and data. This becomes possible due to the small memory footprint of the used software components and the large memory resources of the employed FPGA device. The fastest version of the Nios-II core was selected to achieve the highest possible performance of the TCP/IP stack. The SOC is clocked using a 100 MHz internally generated clock.

The Gigabit Ethernet functionality is realised by a Triple Speed Ethernet (TSE) IP core provided by Altera Corp. The TSE requires the usage of two Scatter-Gather Direct Memory Access (SGDMA) IP cores and is configured to include the 1000BASE-X/SGMII Physical Coding Sub-layer (PCS) function with an embedded Physical Medium Attachment (PMA). This configuration instantiates a High-Speed Transceiver of the Cyclone IV GX device and it allows using direct connection to a Small Form-Factor Pluggable (SFP) module on the BLEDP board. The Management Data Input/Output (MDIO) functionality of the TSE is not included, but instead the SFP modules are configured via the I²C interface realised in the Nios-II software. More information about the TSE configuration can be found in the manufacturer's documentation in [4]. As expected, there are many additional components included in the SOC, e.g. the JTAG UART and the Parallel I/Os, but they are out of the scope of this paper.

Server Process

The embedded software includes mainly ready made components provided by Altera Corp. and other vendors. The main advantage of that solution is the very short time to reach from specifications to the first working prototype. The μ C/OS-II real-time kernel ported for the Nios-II is required by the NicheStack TCP/IP stack. Another advantage of using the provided software libraries is the availability of various services, e.g. the ICMP and DHCP.

Therefore, the only custom software implemented for the BLEDP module is the server application. It was written with the C language as a single-threaded process. The server creates a standard socket which listens for incoming connections. Only one client is allowed to be connected at a time and any additional connection requests from more clients are refused. The server is parsing incoming commands and it disables or enables the requested channels of the BLEDP module. The channels implemented in the user logic are enabled by software using access to the PIO interface included in the SOC. When there are enabled channels, the interrupt coming from the FSM is also enabled by the server. The Interrupt Service Routine (ISR) was im-

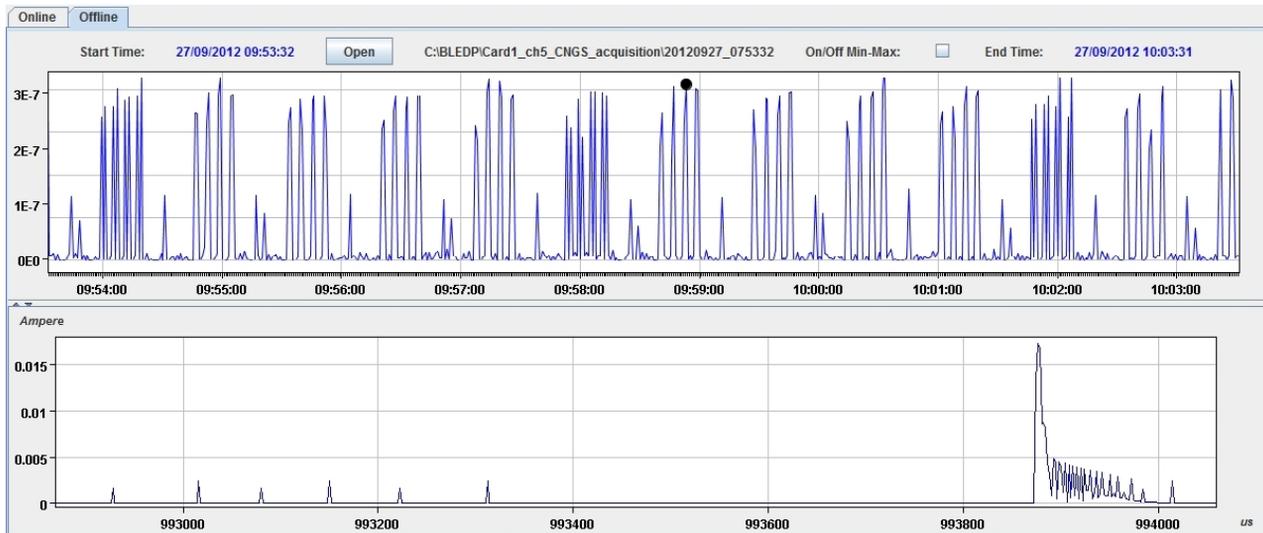


Figure 4: Example of the offline data viewer in the client application.

plemented in the server. The ISR reads a pointer to the BLEDP bundle and it puts it to a global queue. The server reads the content of that queue and it transmits the BLEDP bundles written by the custom logic. In the case of not sufficient network throughput, it is possible that all the buffers allocated will become full with data that were not possible to be transmitted. In that case, the acquisition channel is stopped until the data are transmitted. At the same time, the client receives an alarm.

CLIENT APPLICATION

The BLEDP client has been developed using the JAVA programming language. It is a graphical user interface application, which was built using the Swing framework. Its purpose is to command and collect data from the server as well as to visualise the data collected. The data is plot on the online display and it can be stored in files for offline analysis. The application separates the actions a user can perform providing an online and an offline tab.

Online Display

The data received by the client application are plotted on the online display after some data reduction. An example of the online tab is shown in Fig. 3. A user is asked to select the data type, channel number and other parameters (not shown in the figure). Among them the most important is an “observation period”. In most of the cases only a part of the observation period is visible in the online plot. The visible window is specified by the “window size” parameter.

After setting all these parameters, the user can press the “start” button to command and initialise the data transmission. The data transmission can be stopped at any time, i.e. before the required observation period completes, by pressing the “stop” button. In the online mode, some data reduction is necessary due to relatively high data rate. By default the BLEDP module provides measurements of the

input current with 2 μs period. The client application calculates and displays the average, the maximum and the minimum values observed since the last update. The data reduction is controlled by the user through the “measurement period” parameter.

Offline Analysis

In parallel to the online display, the client can store all the data in files without any data reduction. The BLEDP data is stored with the headers which allows identifying their source, measurement method and continuity problems if any. The offline tab is intended for more accurate data observation and analysis.

The data stored locally in the files is used for the offline data display and analysis. An example of the offline tab is shown in Fig. 4. The folder containing the acquired offline data should be selected by clicking the “open” button. On top of the offline tab, the whole acquisition session is displayed, as soon as the data is selected. This plot is created using statistics read from files which are calculated and stored every second in parallel to the main offline data files. The statistics file contains the average, minimum and maximum acquired values in the current second. These files allow the efficient creation of the acquisition overview over long periods without the necessity of re-examining large amounts of data.

Furthermore, the acquisition session view created serves as the data picker for the detailed analysis. All the points on that plot correspond to consecutive seconds of the acquired offline data. The selected second is indicated by a black dot visible in the figure and, when selected, its complete contents are plotted in the bottom view allowing to perform a very detailed analysis. On that plot the data can be displayed with or without data reduction, converted to different units and zooming is possible in all views.

The examples shown in Fig. 4 is actually data acquired

in the Proton Synchrotron accelerator at CERN with circulating beam. The detailed view of the offline panel in that example is presented without data reduction.

CONCLUSIONS

The TSE IP core is capable to reach the throughput of the Gigabit Ethernet but the software TCP/IP stack running on the Nios-II processor is limiting it to only 20-30 Mbit/s. The TCP protocol has several advantages which justify its usage for the first prototype. Moreover that speed is fully satisfying the needs of the single channel mode which was planned for the first step of the development. For the multichannel mode the UDP protocol will be used to reach the Gigabit Ethernet throughput. For that purpose the custom UDP offload logic will be implemented to be used with the TSE. The multichannel transmission will be used mainly in the Local Area Network (LAN), whereas the single channel can be used also in the Wide Area Networks (WAN).

REFERENCES

- [1] C. Zamantzas, M. Alsdorf, B. Dehning, S. Jackson, M. Kwiatkowski, W. Vigano, "System Architecture for Measuring and Monitoring Beam Losses in the Injector Complex at CERN", 1st International Beam Instrumentation Conference, IBIC2012, Tsukuba, Japan, Oct 1-4, 2012.
- [2] W. Vigano, B. Dehning, E. Effinger, G. Venturini, C. Zamantzas, "Comparison of Three Different Concepts of High Dynamic Range and Dependability Optimised Current Measurement Digitisers for Beam Loss Systems", 1st International Beam Instrumentation Conference, IBIC2012, Tsukuba, Japan, Oct 1-4, 2012.
- [3] M. Kwiatkowski, C. Zamantzas, M. Alsdorf, B. Dehning, W. Vigano, "A Real-Time FPGA Based Algorithm for the Combination of Beam Loss Acquisition Methods Used for Measurement Dynamic Range Expansion", 1st International Beam Instrumentation Conference, IBIC2012, Tsukuba, Japan, Oct 1-4, 2012.
- [4] Altera Corp., "Triple-Speed Ethernet MegaCore, Function User Guide", http://www.altera.com/literature/ug/ug_ethernet.pdf