LOW LATENCY DATA TRANSMISSION IN LLRF SYSTEMS*

D. Makowski, G. Jablonski, P. Predki, A. Napieralski, DMCS, TUL, Lodz, Poland

Abstract

The linear accelerators applied to drive Free Electron Lasers (FELs), such as the X-Ray Free Electron Laser (XFEL), require complex control systems. The Low Level Radio Frequency (LLRF) control system of a linear accelerator should provide a signal to a vector modulator in less than 1 microsecond. Therefore, the latency of communication interfaces is more important than their throughput. This paper discusses the application of serial gigabit links for transmission of data in LLRF systems. The latency of pure serial transmission based on Xilinx RocketIO transceivers was evaluated and compared with Xilinx Aurora protocol. The developed low latency protocol is also presented.

INTRODUCTION

The LLRF (Low Level RF) controller is used to stabilise the electromagnetic field in accelerating cavities of linear accelerators. The LLRF controller calculates the appropriate control signals for the RF source based on the measurements of the RF field in individual cavities to achieve the required RF field stability. The block diagram of a typical LLRF system is presented in Fig. 1.

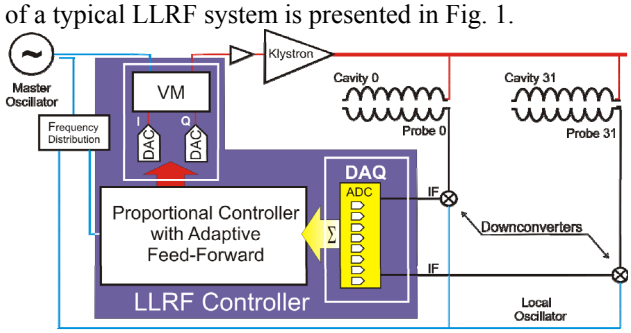

Figure 1: A block diagram of a typical LLRF control system.

The LLRF controller is composed of data acquisition (DAQ) modules, a Computation Unit (CU) and a Vector Modulator (VM). The DAQ modules digitise analogue signals from the downconverters, process I-Q demodulation and calculate the partial vector sum. The result is transmitted to the LLRF CU where a feedback signal is calculated. The evaluated I-Q signal is transmitted to the Vector Modulator module. The VM modifies the Master Oscillator (MO) signal that drives a power amplifier and a klystron. The high power signal is distributed to the cavities through a wave-guide system closing the feedback loop.

DISTRIBUTED LLRF SYSTEM

A single RF station of a LLRF control system of a linear accelerator acquires data from hundreds of analogue channels [1]. Therefore, the LLRF system, designed as a distributed system composed of a few modules, is installed in a shelf. A semi-distributed system, composed of two shelves, is considered for the XFEL accelerator [2].

The connectivity and communication links between distributed subsystems play a crucial role in the control system. The LLRF controller should provide the output signal in less than 1 microsecond to obtain effective acceleration of particles [1]. Since the LLRF controller is composed of a few modules (DAQs, CU and VM) data transmission between processing units should be done within 200 ns. In such an application the deterministic latency of data transmission channel is more important than data throughput.

The flow of signals in the semi-distributed LLRF controller, composed of two computation units (master and slave), is presented in Fig. 2.

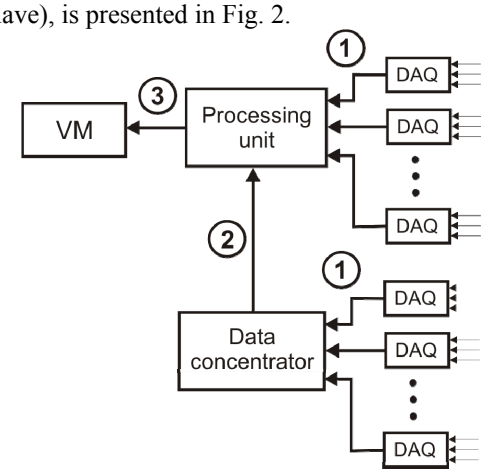

Figure 2: Signal flow in a distributed LLRF controller.

The signals generated by DAQ modules are transmitted using Low Latency Links (LLLs) to the data concentrator and main processing unit (link #1 presented in Fig. 2). The data concentrator (slave unit) receives signals from DAQ modules. The calculated partial vector sum signal is transmitted via LLL #2 to the main processing unit (master system). The processing unit generates a signal that is provided to the VM using LLL #3. Assuming that a single data transfer via LLL connection requires 200 ns, the total latency on connectivity is 600 ns. In such a situation the data processing in the LLRF controller implemented in an FPGA has to finish within 400 ns.

The distributed LLRF control system requires a standard for component housing that assures high reliability and provides suitable links for data transmission.

The xTCA (Advanced and Micro Telecommunication Computing Architecture) standards have attracted the attention of the physics community because they offer various types of data communication channels with high bandwidth, redundancy, high reliability and availability [3]. The standards allow using various types of communication interfaces such as Gigabit Ethernet, InfiniBand, PCI express (PCIe), RapidIO (RIO) and StarFabric [3].

POSSIBLE APPROACHES

A low latency, peer-to-peer communication channel can be realised using many standards. The simplest solution is to apply a parallel bus that allows to send data in a few clock cycles. The latency of such a solution depends on the frequency of the reference clock, bus width and could be in the range of tens of ns (assuming a few hundred MHz clock). However, implementing a high frequency parallel bus could be extremely difficult and expensive in complex distributed systems.

High-speed serial gigabit interfaces are more suitable for data transmission in complex, scalable systems. The easiest way to obtain a high-speed, reliable serial connection is to use the LVDS (Low-Voltage Differential Signalling) standard. Such a connection requires data and clock differential lines. The Double Data Rate (DDR) serial transmission protocol can be used to improve data transmission. The achieved latency of a single link could be around 100 ns for a few hundred MHz clock.

Gigabit Ethernet (GbE), PCI Express (PCIe) and serial RapidIO (sRIO) are among the most popular switched interfaces used nowadays for high-speed data transmission. All of the abovementioned links are supported by xTCA standards.

The GbE standard requires software or hardware implementation of the Ethernet stack. The latency obtained for a software stack solution is relatively high (from several ms to hundreds of ms). The implementation of a hardware stack (e.g. in FPGA) could reduce the latency to tens of microseconds [4]. In real systems the latency on a GbE switch should be also considered. The latency for a pass-through GbE switch is of order of two transceiver delays (ca. 300 ns), for the store-and-forward switch it is increased by the time needed to read the entire frame proportional to its length.

PCI Express, PCIe (Peripheral Component Interconnect Express) has been designed as a computer expansion card standard. PCIe requires a Root Complex (RC) for bus management and configuration. PCIe has a host-centric character, although direct point-to-point communication between the peripherals is also possible. The interface allows to obtain a latency of a few microseconds. The typical latency on a PCIe switch is more than 200 ns.

The serial RapidIO (sRIO) specification is a switched, packet-based technology similar to PCIe. PCIe was initially designed for desktop and high performance computing applications. The sRIO is dedicated for embedded systems applications and does not require a RC. The latency of sRIO is similar to that of PCIe.

Since the switched serial protocols have a relatively high and undeterministic latency, they cannot be used in the LLRF system feedback. One of the potential solution is to use gigabit transceivers available in FPGA devices (e.g. MGT transceivers available in Xilinx T-type devices). Application of a simple transceiver with direct peer-to-peer transmission can significantly reduce latency of the link. Moreover, application of FPGA devices gives the opportunity to simplify the transmission channel and therefore further reduce the latency. Xilinx offer a dedicated low-latency, point-to-point Aurora protocol that can be implemented using MGT transceivers.

Authors carried out a set of measurements using Aurora protocol. Finally, a custom low-latency protocol was designed.

DEVELOPED PROTOCOL

The simplified structure of the Xilinx Virtex 5T gigabit transceiver is presented in Fig. 3.

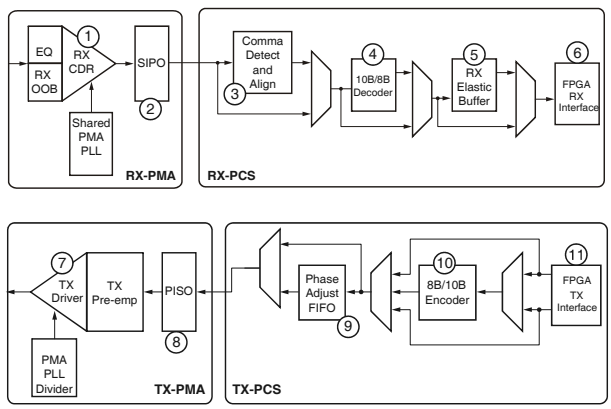

Figure 3: The simplified Xilinx transceiver structure.

In block (1) the data received from the input by the CML receiver is passed through the equalization circuit, compensating for the high-frequency losses in the channel. Then, the recovered clock is reconstructed from the incoming data and the data is sampled at the optimal point. The data is then converted into parallel form in the SIPO (Serial-Input Parallel-Output) block (2). The comma alignment and detection block (3) aligns the stream of bits into the symbol boundaries, using the unique 10-bit symbol with 5 ones in a row (0101111100) called a *comma*. The wire protocol uses 10-bit symbols to ensure a sufficient amount of transitions in the input stream for effective clock recovery and to retain the DC balance needed due to a capacitive coupling. The 10B/8B decoder (4) converts them to the 8-bit form, with 256 different data bytes and 11 control bytes. The receiver elastic buffer (5) allow to transfer the data between the recovered clock domain and the user logic clock domain, with removal or insertion of the clock correction symbols. The minimum amount of data stored in the receiver FIFO for the purpose of symbol rate adjustment between the transmitter and the receiver can be controlled by the MGT parameter CLK_COR_MIN_LAT. From the elastic buffer the data is passed to the 8- or 16-bit FPGA RX interface.

At the transmit path, the data coming from the 8- or 16-bit FPGA TX interface is passed through the 8B/10B encoder. Then the phase adjust FIFO (9) resolves the phase differences between the FPGA clock and transceiver clock. From the FIFO, the data goes to the serializer (8) and the output circuit (7).

For low latency designs, some of the blocks in the receiver and transmitter data path can be disabled.

The TX phase adjust buffer can be replaced by the phase alignment circuit. This mode reduces the latency, but requires extra logic.

The RX elastic buffer can be replaced by the phase alignment circuit. This mode reduces the latency, however the clock recovered from the data input must be used for the FPGA RX interface, which is not desirable in all circumstances. This mode also requires waiting for all the clocks to stabilize and then performing the phase alignment procedure. Therefore it can be rarely used in practice and can only be regarded as a reference value.

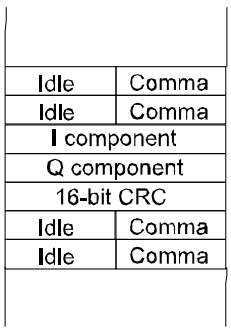

Figure 4: The designed protocol.

The developed protocol is designed to transmit the I and Q data between the input modules and the controller and the controller and the vector modulator. Therefore, the data packet consists of three 16-bit words: the field in-phase component (I), quadrature component (Q) and the 16-bit CRC checksum. Most of the time the link is idle. When the link is idle, the comma character and the idle character are sent alternately to allow bit synchronization and clock synchronization between the transmitter and receiver. The start of frame can be recognized as the first data symbol after a series of control symbols. The designed protocol is summarised in Fig. 4.

EXPERIMENTAL RESULTS

The protocol has been tested on an ML506 Xilinx Virtex 5 evaluation board, equipped with the XC5VSX50T FPGA. The protocol has been tested with the line rate of 3.125 Gb/s. A single SFP module with a 1 m long fiber loopback cable has been used. Four variants of the implementation have been tested:

1. Based on the Xilinx Aurora protocol,
2. Based on direct control of MGT transceivers, with the RX elastic buffer and TX phase adjust FIFO enabled,
3. As above, with the TX phase adjust FIFO disabled,
4. As above, with both RX elastic buffer and phase adjust FIFO disabled.

The Aurora protocol in the default configuration uses the CLK_COR_MIN_LAT parameter equal to 16. To estimate the overhead of the Aurora protocol the latencies for the pure MGT approach were measured for the two values of CLK_COR_MIN_LAT: 4 and 16. The latency was measured using an oscilloscope, between the start of sending the first byte of data in the transmitter and the checksum verification in the receiver. The results of the measurements are presented in Table 1.

Table 1: Latency Measurement Results

Case	1	2	3	2	3	4
MIN_LAT	16	16	16	4	4	N/A
Latency [ns]	185	147	141	109	103	92

CONCLUSION

The commonly used gigabit protocols such as PCIe or Ethernet cannot be used for low-latency communication within the LLRF controller due to the large latency and excessive complexity. A custom protocol has been developed, which allows the latency to be reduced to ca. 100 ns per link. The protocol is secured with a 16-bit checksum to ensure error-free transmission. The obtained results are much better that 350 ns reported for Virtex II Pro in [5]. With the Virtex 5 much more time is left for the controller algorithms. The application of the custom protocol has allowed to reduce the latency by 40 ns in comparison to the Aurora protocol.

ACKNOWLEDGMENT

The research leading to these results has received funding from the Polish National Science Council Grant 642/N-TESLAXFEL/09/2010/0. The authors are scholarship holders of the project entitled “Innovative education ...” supported by European Social Fund.

REFERENCES

[1] D. Makowski, W. Koprek, T. Jezynski, A. Piotrowski, G. Jabłoński, S. Simrock, ”Interfaces and Communication Protocols in ATCA-based LLRF Control Systems”, IEEE TNS 56(4), pp. 2814 – 2820.

[2] D. Makowski et al., “Prototype Real-time ATCA-Based LLRF Control System”, IEEE TNS, (paper submitted).

[3] PICMG, “AdvancedTCA Base Specification,” January 2003, PICMG 3.0.

[4] T. Uchida, “Hardware-based TCP Processor for Gigabit Ethernet” IEEE TNS, 5(3), (June 2008) 1631.

[5] W. Jalmuzna, A. Napieralski, S. Simrock, “Performance of 24 Cavity Vector Sum Controller with Distributed Architecture”, EPAC’08, p. 1401 (2008); wwww.JACoW.org.