A Development of a 40-Gb/s Readout Interface STARE for the AGATA Project

N. Karkour, V. Alaphilippe, J. Collado, N. Dosme, L. Gibelin, V. Gonzalez, X. Grave, J. Jacob, X. Lafay, E. Legay, D. Linget, A. Pullia, M. Quenez, D. Sidler, N. Tessier, and G. Vinther-Jørgensen

Abstract—The Advanced GAmma Tracking Array (AGATA) multidetector spectrometer will provide precise information for the study of the properties of the exotic nuclear matter (very unbalanced proton (Z) and neutron (N) numbers) along proton- and neutron-drip lines and of super-heavy nuclei. This is done using the latest technology of particle accelerators. The AGATA spectrometer consists of 180 high-purity germanium detectors. Each detector is segmented into 38 segments. The very harsh project requirements are to measure gamma-ray energies with very high resolution (\( < 1 \times 10^{-3} \)) at a high detector counting rate (50 K events/s/crystal). This results in a very high data transfer rate per crystal (5–8 Gb/s). The 38 segments are sampled at 100 MHz with 14 bits of resolution. The samples are continuously transferred to the control and processing (CAP) module, which reduces the data rate from 64 to 5 Gb/s. The CAP module also adds continuous monitoring data, which results in total outgoing data rate of 10 Gb/s. The serial transfer acquisition and readout over Ethernet (STARE) module is designed to fit between the CAP module and the computer farm. It will package the data from the CAP module and transmit it to the server farm using a 10-Gb/s user datagram protocol (UDP) connection with a delivery insurance mechanism implemented to ensure that all data are transferred.

Index Terms—Data Communication data transfer, digital integrated circuits, ethernet networks, field programmable gate arrays, high-speed electronics, high-speed networks, nuclear physics instrumentation.

I. INTRODUCTION

ADVANCED GAmma Tracking Array (AGATA) is designed to provide precise information for the study of the properties of the exotic nuclear matter along proton- and neutron-drip lines and of super-heavy nuclei. The AGATA spectrometer consists of 180 high-purity germanium detectors. The AGATA project [1] instrumentation has been updated through three phases (Phase0, Phase1, and Phase2). This actual instrumentation is part of the phase 2 electronics. Fig. 1 shows the different phases of AGATA from Legnaro (phase 0) to GANIL (phase 1). Each detector is segmented into 38 segments. Each segment is preamplified and the output is sampled at 100 MHz with 14 bits of resolution using flash analog-to-digital converter (FADC) boards called the Digitizers with optical fibers outputs (DIGIOPT12) [8]. The total raw data rate per detector is around 64 Gb/s (1.6 Gb/s \times 38 segments plus additional diagnostic and identification information of 2 \times 1.6 Gb/s). The raw data are processed in real time, compressed to 10 Gb/s, and sent to the serial transfer acquisition and readout over the Ethernet (STARE) module. The general overview of the AGATA detector acquisition is shown in Fig. 2. In this article, we will report on the STARE prototype hardware and firmware design. We will describe the research and development phase for the proof of concept (POC). This includes hardware design, firmware design, the results, and a description of the future work.

Fig. 1. AGATA phases 0, 1, and 2 detector design.

Fig. 2. Instrumentation of an AGATA detector. PACE block diagram and specifications.
The STARE module design is an FPGA Mezzanine Card (FMC) standard module, which will be mounted on a preprocessing and control through the Ethernet (PACE) module. The PACE [4] module is a collaboration between the IFIC (Valencia, Spain), the Department of Electronic Engineering (Valencia, Spain), CNRS/IN2P3/IJCLAB, and INFN Milan.

The construction of STARE is fully financed by the Institut National de Physique Nucléaire et de Physique de Particule (IN2P3) of the Centre National de Recherche Scientifique (CNRS) in France through the AGATA Memorandum of Understanding (MOU).

II. RESEARCH AND DEVELOPMENT PRELIMINARY PHASE

The research and development phase started in 2017, looking for a 10-Gb/s Ethernet IP design compatible with Xilinx FPGAs. After gathering and sourcing different solutions (costs of up to 150 k€ from a third-party company, which is extremely expensive and not possible with the financial plan for the AGATA project), a collaboration was established with Zurich University (ZTH) to integrate a 10-Gb/s IP developed at the university with the VC709 Xilinx evaluation board. This IP [2] is an open-source IP available to all research institutions and is developed by Sidler [6] who kindly helped with the integration. The first validation was based on a simple data generator IP that could transfer data at various transfer rates, and on the server side, the Internet Performance Working Group (IPERF) [7] program was used to receive the data from the VC709 10-Gb/s user datagram protocol (UDP) port. This program helped us to validate the IP design with the AGATA data transfer constraints. The first results in Fig. 3 are output by the IPERF program. They show that the transfer rate can reach 8.6 Gb/s from the VC709 to the computer server. Three types of tests were made: 1, 5, and 10 Gb/s. The IPERF output has five columns: the first column is the interval time, which is the test period, and here, it was chosen to be 5 s of continuous data transfer. The second column is the total data transferred during each test. The third column is the measured network bandwidth transfer rate (Gb/s). The fourth column is the network jitter, and the fifth column is the percentage of data loss over total data transferred. The results showed that at 1 Gb/s, the data loss is <0.5%, at 5 Gb/s, it is <1%, and at 10 Gb/s, the transfer rate is between 8 and 8.6 Gb/s with a high data loss of <20%. It is important to point out that the data loss requirement in AGATA is 0%, and the receiver server bandwidth is not fixed and depends on its data rate capacity while doing other tasks. It was decided during this research and development phase that the STARE must have multiple UDP transceivers to fulfill the AGATA design requirements. It is important to mention that there is no software optimization on the receiver-side server and this plays an important role in data rate transmission. Moreover, the Virtex 7 FPGA is a small FPGA to house the four UDP transceivers and the high-speed clock connection used for the transceivers is not flexible. This test convinced us to proceed with the research and development phase while using a more powerful FPGA family. The design of the POC will be using the KU040 FPGA integrated in the KCU105 development board.

III. PROOF OF CONCEPT

The POC was made to validate the complete instrumentation chain of one AGATA channel. It is composed of using one AGATA DigiOpt12 module (digitizer data), the Instituto de Física Corpuscular Universitat de Valencia (IFIC-UV) setup simulating the preprocessing research and development. It uses the Zync Ultrascale TE0808 system on module (SOM) mounted on a TEBF0808 evaluation board (to record traces from the DigiOpt12 and send them to the STARE), and the CSNSM setup composed of IJCLAB STARE research and development evaluation board the KCU105 with an optical fiber adaptor module (FMC with SFP modules). The server STARE01 is used to receive the STARE data. Fig. 4 shows the POC setup block diagram. Fig. 5 shows the POC modules all together installed in IJCLAB.

The STARE firmware design was based on the following characteristics:

1) UDP IP provided by David Sidler (ETH Zurich) [2] and IPBus firmware from CERN [2];
2) event splicing/reconstruction;
3) one UDP transceiver out of four programmed;
4) package loss simulation.
There were several tests made to validate the AGATA POC. The first test was to check the STARE UDP transmission using a data generator implemented in the STARE firmware. The second test was to connect the IFIC evaluation board to the STARE without the DIGIOPT module and transfer data from a data generator implemented in the IFIC board firmware. The third test was to connect all modules. The tests were made successfully between the three modules. The server software used to validate the data quality was only made to work for the first and second tests. Moreover, two types of receiving data programs were used on each test. The first program was receiving data but not writing to disk. The data in Fig. 6 show the results from this test. With a transfer rate at 5 Gb/s and an event size of 2 kB, there is 0% data loss. As the data sent from the data generator followed predefined pattern, it was possible to verify the received data on the server. This is how it was possible to verify that no data were lost. The table shown in Fig. 6 only shows data for a small part of a test, which lasts for more than 1 h. It was not run continuously for 24 h as it was clear that the data loss was 0%. This test was done every day for about one week with a duration of 1 h to make sure that the system was stable over time. The second program used was receiving data and writing to disk. The data gathered from this test is shown in Fig. 7. Here, it can be seen how the loss rate rises as the program was not able to store the data as quickly as it arrived. This test can only run for less than a second before overloading the server. This test confirmed that either a more optimized software, the use of multiple 10-Gb/s UDP ports to multiple servers or one UDP port to multiple servers is required. Another idea that arose from this test was to develop an IP guaranteeing reliable data transfer at 10 Gb/s. This will be done in future firmware upgrades.

IV. STARE HARDWARE DESIGN

A. STARE Definition

The STARE is part of the preprocessing electronics of the phase 2 electronics for the AGATA project. It is an FMC board that will be mounted on the PACE module. The AGATA germanium detector signals are digitized in the DIGIOPT modules and the continuous data are processed inside the PACE module. The Readout manager of the PACE module is based on the Xilinx AXI stream AURORA interface. It sends data to the STARE and the STARE processes the received data at up to 10 Gb/s in order to transfer it to the server farm using the UDP protocol. The STARE module is equipped with external memory to allow retransmission of any packages not correctly received by the server farm. Fig. 8 shows the STARE conceptual design and philosophy of work. It can handle up to four independent 10-Gb/s transceiver lines on its input and send data to four or more independent 10-Gb/s servers or computer farm. One UDP channel can send data to different servers, and the STARE has four independent UDP channels.

B. STARE Description

The STARE board is composed of an SOM, an FMC connector, and four SFP+ 10 Gb transceivers. It contains the following components: two EEPROMs, a Joint Test Action Group (JTAG) interface, an external power input, a high-precision oscillator, and clock generator. Fig. 9 shows the STARE block diagram. The STARE design respects all electrical requirements described in the FMC Vita standard [9]. It does not fully comply with the mechanical requirements in the Vita FMC standard because of the SOM and the four SFP+ connectors size, which makes the height of the STARE over 10 mm that is the requirement stated in the Vita FMC standard. The length of the STARE board will also be longer than the VITA recommended rules. The PCB design will have a rugged metallic border to ensure thermal conductivity since the SOM and SFP transceivers’ temperature might increase during full-speed transmission.

C. SOM Module

The SOM that will be used on the STARE module is the TE0841 TRM (SOM) from Trenz. Fig. 10 shows the SOM
module picture. It requires $50 \times 40$ mm of space and $50$ mm height. The SOM is connected through two LSHM-150-04.0-L-DV-A-S-K-TR (100 pins) connectors and one LSHM-130-04.0-L-DV-A-S-K-TR (60 pins) connector to the STARE. The SOM module is composed of a KU040 Xilinx FPGA, 2 GB of external DDR memory used to write event data and read them back to ensure that lost data can be retransmitted again, a programmable clock generator/phase locked loop (PLL) to distribute up to four independent clocks, and a system monitoring block controlled by a complex programmable logic device (CPLD) to control power supplies voltages and FPGA temperatures.

D. FMC Interface

The STARE is mounted on the PACE board through an FMC high pin count (HPC) connector. This connector will contain four 10-Gb/s transceiver data pins, which will be connected to one of the FPGAs transceiver banks. The GBT-M2C clk0 clock pins will be connected to the 156.25-MHz clock output of the 5338 programmable PLL to propagate the clock signal from the STARE board to the carrier card.

The FMC standard requires a single connector board to have a fixed width of 69 mm. The HPC connector needs a space of $14.48 \times 53.90$ mm. Two holes will be mounted on each side of the FMC connector to fix the STARE on the PACE board through spacers so that the FMC connector does not support the mechanical stress of the STARE. There are 20 low voltage differential signaling (LVDS) pair signals connected between the STARE SOM and the PACE FPGA for future use (such as the IPBus interface or any other control lines). The clk0-M2C-CC signal will connect a clock signal from the SOM to the PACE FPGA. All power and power control pins respect the VITA standard. A set of Reset signals is connected to manage hardware and software resets on the STARE. These will be used for initialization procedures. Because the STARE board contains several high-speed computing technologies and programmable components, the power supply sequence and the initialization procedure are very complex. The high-speed links and data management are extremely complex in the case of the AGATA instrumentation system and the data acquisition software. Any desynchronization in the system clock, memory buffers, system counters, or online preprocessing steps will lead to data recording errors and system hangout.

E. Prototype Board Design

The STARE board prototype has some functions, which will not be used in the production board. The power supply management on the STARE board is very complex and mandatory because the power up sequence of the SOM module and the SFP+ transceivers is a challenge. This is because the carrier boards used to test the STARE board might not deliver the necessary power supply at the first power-up stage of the STARE prototype. Fig. 11 shows the block diagram of the STARE power supply management and power-up sequence. Moreover, the JTAG programming will use different modes of connections (FMC JTAG, simple Xilinx HX Programmer, or a JTAG SMT2 module). In the production phase, only the FMC JTAG programming will be implemented. This extra functionality will make the STARE prototype board longer than the production board but will make development and debugging significantly easier. Fig. 12 shows the STARE prototype layout.

F. Production Board Design

The STARE production board will be smaller than the prototype because all onboard JTAG connections and the power supply multiplexers will be removed. Fig. 13 shows the production version of the STARE board layout.
started using the VC709 Virtex 7 technology from Xilinx. The first firmware was developed inside the VC709 to validate the conceptual design of the STARE, using 10 Gb/s transfer results over UDP FPGA Network Stack developed by David Sidler from ETH Zurich. Virtex 7 showed immediately its technology limitations to fulfill the AGATA project requirements. The main problem was a lack of flexibility regarding the transceiver clocks. The POC firmware was developed using the KCU105 development board sold by Xilinx. This development board is based upon a modern Kintex Ultrascale (KU040) FPGA and hosts a number of connectors used for high-speed data transfer to and from the FPGA.

The POC firmware was very simple and it consists of a dummy data generator, an IPBus interface firmware from CERN [2], and the FPGA network stack developed by Sidler et al. [3] from ETH Zurich. These three objects were combined into a single project. The IPBus is providing slow control and debugging functionality via a 1-Gb/s Ethernet connection, while the FPGA network stack provides a 10-Gb/s Ethernet link supporting UDP [6].

The merging of these two code bases resulted in a fairly complex project. As the FPGA network stack and IPBus had to share FPGA clocking primitives for their high-speed transceivers, a good code structure was not initially achieved. This resulted in compiled very high descriptive language (VHDL) files and a project that was hard to make sense of.

To make the project more accessible, the structure shown in Fig. 14 was developed. Based on the experiences from the initial implementation, a number of improvements could provide a much cleaner code structure making the project easier to manage. While restructuring the code, the IPBus slow control functionality should also be implemented. As shown in Fig. 14, the STARE firmware is divided into four modules: the Aurora interface, the data manager, the IPBus interface, and the UDP interface [5]. Each module contains an IPBus slave providing slow control and status monitoring. The Aurora interface receives data from the PACE module FPGA through gigabit transceivers. These transceivers use a 156.25-MHz clock to achieve 10 Gb/s. The data manager takes the event data received and splice it into small packets to be sent over UDP. The UDP interface adds specific protocol data to each packet and transmits the packet to the server.

A. Reliable UDP 10-Gb/s Transfer Selective Repeat Protocol (SRP)

One of the requirements for the STARE board is that all the data received from the preprocessing must be delivered to the server. As the transport protocol used is UDP, this functionality is not guaranteed by default and must therefore be added. The other widely used transport protocol TCP does provide guaranteed delivery. Thus, an explanation for why UDP was chosen anyway is needed. The FPGA network stack provided by Siddler et al. [3] did originally contain an implementation of both UDP and TCP. Both interfaces were tested extensively by another visitor (A. Triossi) in the early phase of the STARE research and development phase. Here, it was discovered that the TCP implementation could not provide more than about 2 Gb/s of bandwidth on a single TCP session. The reason for this can be found in the documentation for the FPGA network stack [3]. Here, the stated goal is to develop a TCP implementation with the capability of handling several TCP sessions at the same time efficiently. This effectively means that the TCP implementation is optimized for a completely different use case than that of the STARE board. The STARE board will only have one session between the board and a server and would therefore never utilize the full 10-Gb/s bandwidth. This leaves the UDP interface as the only feasible option. The UDP interface is able to utilize the full bandwidth for a single session. However, the downside to this is that a protocol is needed on top of the UDP protocol to guarantee delivery. This will slightly decrease the available bandwidth and require time to be implemented. Implementing a reliability mechanism for UDP is something that has been done before. In cases where TCP is too resource heavy to implement, a UDP connection with a simple reliability protocol on top can often be the solution. When deciding on a protocol, it is important to define what exactly is expected. For the STARE project, the requirement is that all data are delivered and delivered exactly once. This means that lost data should be resent and data that are sent twice should be discarded. Several protocols were studied and the chosen protocol was the SRP, which is the more efficient option. Fig. 15 shows the block diagram of the SRP.

This, however, comes at the cost of increased complexity. The operation of the protocol is shown in Fig. 16. When using the SRP, only the lost frame will have to be sent back.
Fig. 16. SRP mechanism with intelligent handshake process, to minimize time shift, and synchronizes quickly.

This results in better utilization of the available bandwidth, especially if the loss rate is high.

It also means that the memory interface must allow single frames to be readback for retransmission. As each frame is treated individually, it also means that a timeout mechanism for each frame is needed. This increases the implementation complexity on the sending side.

The core of the SRP implementation is the RUDP core. RUDP is short for reliable UDP, which is what the SRP provides. The RUDP core keeps track of all the active frames and generates timeouts used to resend frames in case an acknowledgment is not received.

When a frame is allowed to pass through the data stopper, the frame is marked as active in the RUDP core. The next step in the frame lifecycle is for the timeout countdown to start. This happens when the frame has passed through the header extraction module. In this way, the timeout will not be affected by frames readback from the memory delaying the frame transmission.

The timer is now running. If the acknowledgment frame is received before the timeout occurs, the frame is marked as inactive and the RUDP core is finished with the frame. If the acknowledgment frame is not received in time, the frame times out. When this happens, a request is sent to the memory interface block triggering the frame to be read from memory. When the frame passes through the header extraction module, the timeout is started again. This process will continue until the acknowledgment frame is received. The flow described is shown in Fig. 17.

B. Memory Interface

The memory interface is a complex block designed in the STARE firmware. The reason for the complexity is that it must be possible to readback frames for retransmissions while still allowing the incoming data to be stored at up to 10 Gb/s. The POC firmware can only store data from a single 10-Gb/s input. With the Trenz SOM module, it might be possible to handle two 10-Gb/s inputs depending on the network latency and memory bandwidth. The memory management block diagram is shown in Fig. 18.

VI. RESULTS

The STARE hardware computer-aided design (CAD) is finished, and the prototype board is still under production. All the hardware tests were done with the KCU105 evaluation board. The real prototype tests will be done starting Q1 2021.

The hardware tests will include IBERT tests to check for the eye diagram of the STARE PCB. The KCU105 hardware was already qualified by Xilinx. There was no need for further verification. The implementation of the UDP interface was a success for the project and looks very promising.

The implementation of the SRP proved more complex than expected. The amount of code and complexity required for solving this simple task was surprising. The result is, however, a nicely structured and efficient implementation. There are several possible improvements and optimizations for the future. More rigorous testing and qualification is also needed.

The implementation around the RUDP core has also proved to be smart. As the RUDP core has all the information about each frame and the status of the window overall, more
functionality can be added. An example could be a mechanism for stopping data transfer if the loss rate becomes a real issue.

The adaptation of Vivado HLS for parts of the implementation has been a big success. Compared to the experience of developing traditional VHDL Vivado, HLS is much more efficient for implementing many of the modules needed for the AGATA project. The possibility of testing the code with C++ test benches also helped discover and remove bugs before the module was implemented on the FPGA.

The implemented server software is in no way optimized. It is proved that the data loss comes from the server receiver that cannot handle the UDP transfer when it is loaded by other tasks or if the full bandwidth of the 10-Gb/s connection is used. The connection between the KCU105 and server did not go through any switches or other hardware, meaning that any problems were caused by either the KCU105 or the server.

It was proved that if the server was tasked with saving the received data, then the data loss percentage rises immediately. It is therefore crucial that the software on the receiver server must be optimized, and this task is programmed in 2021. This includes an optimization of the way memory is handled and the implementation of checks on the whole data frame and not just the header. This will make it possible to use the software for the final qualification of the firmware.

Results from the test of the RUDP protocol implementation can be seen in Fig. 19. The data shown in the table are taken from the server software. This shows a test where the KCU105 was sending data at 5 Gb/s with no simulated loss. The event size was set to 8192 B and each UDP packet contained 1400 B of data. This clearly shows that the implementation functions and can transfer data at high speed.

Attempts were also made to increase the bandwidth further. Unfortunately, the implemented data generator has no steps between 5 and 10 Gb/s. When tested at 10 Gb/s, the software only reported a total transfer rate of around 6 Gb/s. At the same time, the monitors in the firmware reported that the full link capacity was used. Based on the testing carried out, it is fair to assume that the issue is caused by the server being unable to process the incoming data.

The design used in the results above is composed of 1 × 10 Gb/s UDP interface, which can handle one input link with one RUDP core implemented. The design used 20% of the FPGA resources. Another design was implemented using two independent 10-Gb/s UDP interfaces without the RUDP core, and the design used 30% of the FPGA resources.

These designs were implemented on the KCU105 evaluation board. The STARE prototype was not ready for the tests with the SOM module mounted on that. It is known that the DDR memory bandwidth on the STARE SOM can only handle two UDP links using the RUDP protocol. The remaining two links can be utilized for 10-Gb/s UDP connections but without guaranteed delivery. Using all four links will result in a complex design. However, as shown above, there should be enough FPGA resources for enabling all four links.

VII. CONCLUSION

The STARE hardware prototype is under production, and ten modules will be delivered Q4 2020 with qualification milestone with the PACE module in Q1 2021 and with the AGATA detector Q2 2021. Production will be delivered in Q4 2021. The firmware will be implemented inside the SOM modules using the same code as the KCU105 FPGA since both FPGAs are the same and the external DDR memory interface is the same. The firmware will be qualified in Q1 2021 and the upgrades will continue until Q2 2022.

ACKNOWLEDGMENT

The authors would like to thank all the people involved in this project and their financial department institutes who supported their research and development (IN2P3). The authors would like to thank first David Sidler from the ZTH Zurich (now he is working at Microsoft Company) without him they would not validate the preliminary research and development phase. They would also like to thank Andrea Trissi for the great work he has done to help them progress in their research and development phase. They would like to thank two excellent students from ESME Sudria University, Melissa Quenez, and Nicolas Tessier. They would like to give great thanks to Gustav their student and friend who put the necessary ingredients to make the POC possible and who designed the SRP. His contribution is extremely valuable. They thank all their technical staff who has invested a lot of time and effort in this project and the scientific staff, who helped them to achieve their goals in this project. Many thanks to all the AGATA collaborators, especially their Valencia Team and Milano Team who design the PACE module.

REFERENCES

[1] S. Akkoyun et al., “Agata—Advanced gamma tracking array,” Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip., vol. 668, pp. 26–58, Mar. 2012.
[2] C. G. Larrera et al., “IPvbus: A flexible Ethernet-based control system for xTCA hardware,” J. Instrum., vol. 10, no. 2, Feb. 2015, Art. no. C02019.
[3] D. Sidler et al., “Scalable 10 Gbps TCP/IP stack architecture for reconfigurable hardware,” in Proc. IEEE 23rd Annu. Int. Symp. Field-Program. Custom Comput. Mach., May 2015, pp. 36–43.
[4] J. Collado et al., “A new preprocessing and control board for the phase 2 electronics of AGATA experiment,” in Proc. IEEE-NPSS Real Time Conf. (RT), Padua, Italy, Jun. 2016, pp. 1–4, doi: 10.1109/RTC.2016.7543161.
[5] G. Vinther-Jørgensen, “Internship report 2,” Dept. Elect. Eng., Tech. Univ. Denmark, Lyngby, Denmark, Tech. Rep., 2019.
[6] D. Sidler, “In-network data processing using FPGAs,” Ph.D. dissertation, Dept. Comput. Sci., ETH Zürich, Zürich, Switzerland, 2019.
[7] L. Cottrell, “Measuring end-to-end bandwidth with Iperf using Web100,” in Proc. Passive Act. Meas. Workshop, 2003, doi: 10.2172/813039.
[8] A. Pullia, D. Barrientos, and S. Capra, “Open-source diagnostic tool with GUI for the new AGATA/GALILEO/EUCLIDes digitizer cards,” in Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC), Seattle, WA, USA, Nov. 2014, pp. 1–5, doi: 10.1109/NSSMIC.2014.7431138.
[9] ANSI/VITA 57.1 FPGA Mezzanine Card (FMC) Standard, VMEbus Int. Trade Assoc., Fountain Hills, AZ, USA. [Online]. Available: http://www.vita.com