

24TH INTERNATIONAL WORKSHOP ON RADIATION IMAGING DETECTORS
OSLO, NORWAY
25–29 JUNE 2023

System for the fast readout and tests of pixel IC operating in single photon counting mode using PCIe-based FPGA

P. Skrzypiec^{ORCID},* P. Gryboś^{ORCID} and R. Szczygieł^{ORCID}

*Department of Measurement and Electronics, AGH University of Krakow,
al. Mickiewicza 30, 30-059 Krakow, Poland*

E-mail: pawel.skrzypiec@agh.edu.pl

ABSTRACT: Hybrid Pixel Detectors (HPDs) have become popular in particle and photon detection techniques in recent years. This type of devices consists of two parts: a pixelated sensor (based on Si, Ge, GaAs, CZT, etc.), and a readout Integrated Circuit (IC), which usually contains thousands of pixels and millions of transistors. ICs suffer from the inaccuracies of manufacturing processes, therefore HPDs have to be thoroughly tested before the sensor bump-bonding process.

This paper presents a highly efficient system for the automated testing of pixelated HPDs. The presented solution is based on the Intel Arria 10 GX Field Programmable Gate Array (FPGA) development kit and a Linux-powered Personal Computer (PC), connected via Peripheral Component Interconnect Express (PCIe) 8x Gen. 3 interface. The proposed system has been built of well-thought-out modules connected through the set of precisely defined interconnects. This approach enabled the development of an architecture that may be easily implemented in both PCIe-based systems and System-on-Chip devices, such as Intel Agilex SoC. The presented system has been tested with both a manufactured IC and a model implemented in the FPGA.

KEYWORDS: Detector control systems (detector and experiment monitoring and slow-control systems, architecture, hardware, algorithms, databases); Digital electronic circuits; Hybrid detectors; X-ray detectors

*Corresponding author.

Contents

1	Introduction	1
2	Hardware architecture	2
2.1	Tested chip	2
2.2	Transceiver	3
2.3	Sequencer	3
2.4	Data processing unit	4
2.5	Data storage controller	4
3	Software architecture	5
3.1	Kernel driver	5
3.1.1	Initialization	5
3.1.2	Userspace interface	6
3.2	Shared libraries	6
3.3	Interactive system controller	6
3.4	Non-interactive system tester	6
4	Test results	7
5	Comparison with existing solutions	7
6	Conclusions	8

1 Introduction

Hybrid Pixel Detectors (HPDs) are Application-Specific Integrated Circuits (ASICs) and are commonly used in many industries, high-energy physics experiments, and medicine [1]. These devices consist of two main components: pixelated sensors and readout Integrated Circuits (ICs), which are manufactured separately and bump-bonded during an integration stage.

Frequently, semiconductor readout systems have to be compatible with sensors that contain thousands of pixels. Therefore, the devices are made up of millions of transistors, each of which may suffer from inaccuracies in manufacturing processes, such as the spread of component parameters dependent on the location of those elements and environmental conditions. It implies that the bump-bonding of a sensor and readout electronics must be preceded by the reliable and meticulous verification of the manufactured IC.

So far many approaches to HPDs testing have been presented. One of the most interesting examples is the integration of pixel matrices with microprocessors [2], and the verification of readout systems performed using embedded processing units. Unfortunately, sometimes project requirements prevent the embedding of microprocessors inside readout circuits. In those scenarios, some assistive devices, such as Field Programmable Gate Arrays (FPGAs) [3] or National Instruments PXI platforms [4], must be used to control and test an HPD.

The authors propose the construction of Peripheral Component Interconnect Express (PCIe)-based environments for HPD testing and control, in which responsibilities are distributed among software executed by x86 Central Processing Unit (CPU) and hardware components implemented in an FPGA. Solutions built of Linux-powered Personal Computers (PCs) and FPGAs connected via PCIe interface are suitable for detectors testing and control, because they provide configurability, flexibility and high-bandwidth inter-device data transmission. Additionally, the implementation of device models inside FPGAs may significantly reduce ASICs testing times by enabling simultaneous chip design and test environment construction.

The article is organized as follows. Sections 2 and 3 present the hardware and software architectures of the proposed solution, respectively, section 4 presents the results of conducted experiments, and section 5 contains conclusions and suggestions for further research.

2 Hardware architecture

The constructed system consists of a Linux-controlled PC and an Intel Arria 10 GX FPGA development kit. These devices communicate with each other through the PCIe 8x Gen. 3 interface, which allows data to be exchanged with a throughput of up to ~ 7.8 GBps. An inter-device data transmission is based on two mechanisms: memory-mapped write/read operations and PCIe Direct Memory Access (DMA) transactions. The former one is used for non-time-critical operations, such as a status register readout, and the latter one is used for a high-speed detector data transmission from an FPGA On-Chip Memory (OCM) to a CPU Double Data Rate (DDR) memory.

The configuration of the FPGA PCIe interface is performed by Intel Arria 10/Cyclone 10 Hard IP for PCI Express provided by Intel® Quartus® Prime Pro. This Intellectual Property (IP) core not only configures edge connector transceivers but also processes all transactions initiated by the CPU. It performs all necessary address translations required for the enablement of the software-hardware interfacing, such as custom IP registers accessing. The overall architecture of the designed system is shown in figure 1.

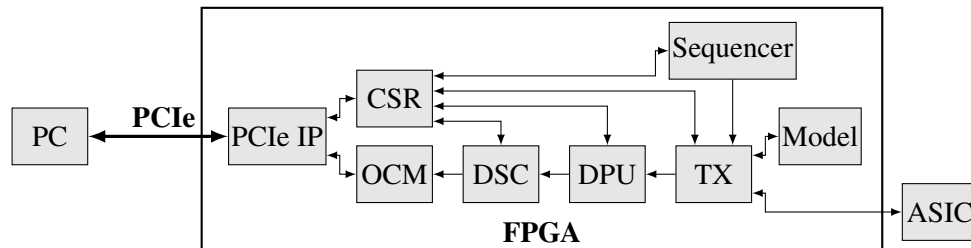


Figure 1. The overall architecture of the developed system (Control and Status Register (CSR), Data Processing Unit (DPU), Data Storage Controller (DSC), Transceiver (TX)) with the connections between components.

2.1 Tested chip

The constructed system is adapted to test a readout IC of pixel architecture, designed for CdTe pixel detectors used in X-ray imaging applications (for the energy range 20–140 keV) with moving objects [5]. The IC core is a matrix of 192×64 square-shaped pixels of $100 \mu\text{m}$ pitch operating in a single photon counting mode. Each pixel contains a fast analog front-end followed by 3 independently

working discriminators and 3 ripple counters. Such a pixel architecture allows photon processing one by one and selecting the X-ray photons according to their energy. The peripheral area located at the bottom of the integrated circuit contains a bandgap reference source, bias Digital to Analog Converters (DACs), an Input/Output (I/O) control logic, a slow control for setting the configuration register in each pixel, Low-Voltage Differential Signaling (LVDS) drivers and receivers.

The IC communicates with the constructed system through three LVDS lines: clock, data in, and data out. The behavior of this device is controlled by a simple protocol that allows for, e.g., updating pixels local configurations. A block scheme of the tested chip is shown in figure 2.

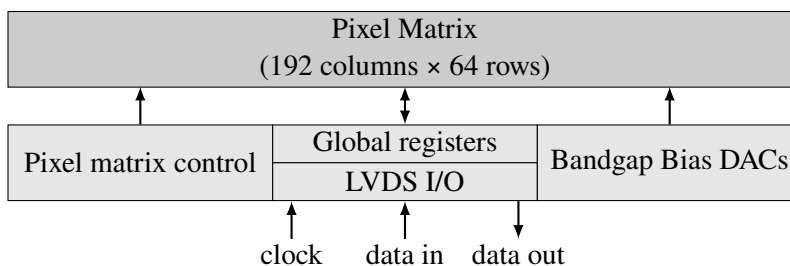


Figure 2. The block scheme of the tested IC with control and data flow.

2.2 Transceiver

Transceiver (TX) is a dedicated component responsible for exchanging data between the FPGA and the tested IC. Its signal lines may be connected to the manufactured chip or the Hardware Description Language (HDL) chip model implemented in the FPGA, as shown in figure 3. The transceiver consists of the following subcomponents:

- clock generator; this module generates a signal that synchronizes flip-flops implemented inside the tested chip and inter-device data transmission. The frequency of the generated clocking signal can be dynamically changed by updating the one of the transceiver CSRs.
- transmitter; it serializes and transmits data to the tested IC. Transmitter supports the transmission of variable-length payloads and, as a result, the serialization of commands of different sizes. It manages the operation of the clock generator and ensures that the clock is generated only during data transmission, which prevents data overwriting, decreases the tested chip power consumption, and reduces the impact of noise on the radiation registration results.
- receiver; this module deserializes data received from the IC. It is implemented as a 192-bit shift register, the content of which is updated on the rising edge of the clock generated by the clock generator. The receiver monitors the number of bits acquired and synchronizes the received data processing by generating a signal that confirms the validity of the data.

2.3 Sequencer

The transceiver can be directly controlled by software by interacting with its CSRs. This mechanism enables full control over the tested IC, but due to the non-time-deterministic code execution, may be insufficient in precisely time-restricted experiments. To enable transceiver control with a precision of one clock cycle, a dedicated execution unit has been implemented.

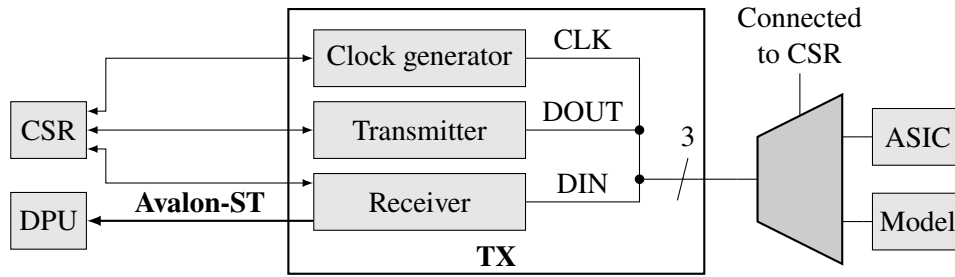


Figure 3. The components involved in communication between the test environment and the tested IC and its model (Avalon Streaming Interface (Avalon-ST), Clock (CLK), Data In (DIN), Data Out (DOUT)).

The sequencer is a programmable execution unit, which may be used to control the transceiver, and consequently the tested IC, with a precision of one clock cycle. This component operates by reading data from a dedicated Random-Access Memory (RAM) and delivering them to the transceiver. Its functionality is not only limited to data forwarding, but it is also capable of detecting some predefined special symbols, such as an idle command that delays two subsequent transceiver interactions and, as a result, delays the transmission of tested IC commands. The overall architecture of the sequencer is shown in figure 4.

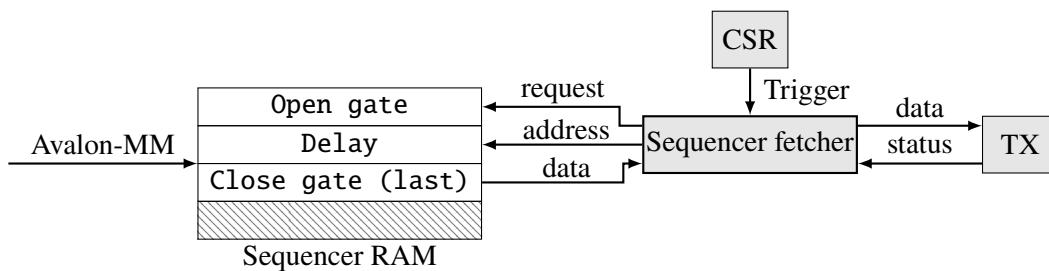


Figure 4. The storage and transmission of control sequences (Avalon Memory Mapped Interface (Avalon-MM)).

2.4 Data processing unit

The Data Processing Unit (DPU) is a data processing module whose functionality is distributed among multiple Processing Engines (PEs), each of which performs a precisely defined function, such as data buffering or a format conversion. PEs are connected to each other via Avalon-ST interfaces and may be dynamically attached or detached from a processing pipeline by bypassing as shown in figure 5. A new data processing function may be easily added to the DPU by implementing a new PE and its placement in the pipeline. Data streams connected to the presented decoder differ in size because this module converts raw data read from the tested IC (192 raw bits) to the format in which the subsequent memory words contain the contents of the subsequent pixels (192 16-bit pixels).

2.5 Data storage controller

The Data Storage Controller (DSC) is a component responsible for data storing and preventing data corruption. It receives data from the DPU via the Avalon-ST interface and stores them in the FPGA OCM using Avalon-MM bus transactions. Its CSR contains read and write pointers that are used by

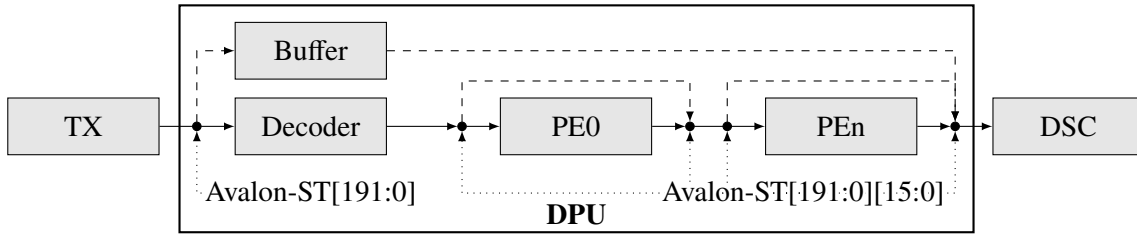


Figure 5. The internal architecture of the DPU — the processing pipeline consists of dedicated PEs.

the module to determine which memory ranges have been read by software. This approach ensures that stored data will not be overwritten and enables using of the OCM as a First-In First-Out (FIFO) buffer.

During operation, the DSC increases the write pointer after each Avalon-MM transaction completion. Software monitors this pointer and performs the OCM readout if its value changes. Each of the OCM readouts is followed by the read pointer update. Both pointers are compared by the DSC to determine whether an Avalon-MM store transaction will not corrupt data unread by the software. If there is not enough space to store pending data in the OCM, the DSC sets the overflow flag in its CSR. The overall architecture of the DSC is shown in figure 6.

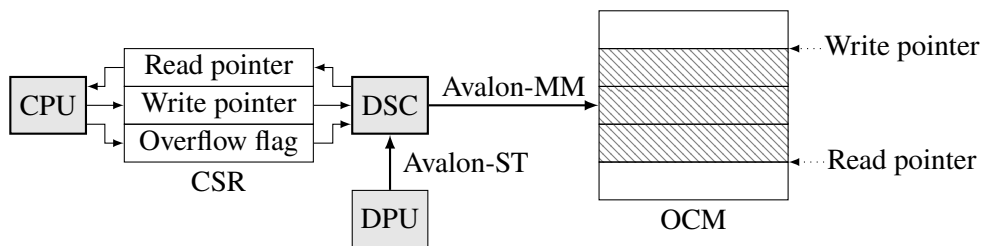


Figure 6. The solution used to store and provide to the PC data received from the tested IC.

3 Software architecture

The developed software is managed by Linux 6.4.13 released for Fedora Workstation 37. Intel(R) Core(TM) i7-11700K CPU with 64 GB DDR4 memory and 2 TB Solid State Drive (SSD) hard drive ensure that experiments are performed efficiently and reliably. The software consists of four components: kernel driver, shared library, interactive system controller, and non-interactive tester.

3.1 Kernel driver

3.1.1 Initialization

The developed kernel driver is implemented as a loadable kernel module. During its loading, the list of supported devices is compared with the list of detected PCIe devices. If a match occurs, the following initialization sequence is executed:

- device-managed memory allocation; the `devm_kzalloc` function is invoked to allocate a memory for a driver internal data; this function is used instead of the `kmalloc` to transfer the responsibility of the allocated memory freeing from the user to the kernel,

- PCIe device initialization; the device internal resources are initialized and requested using the PCIe Application Programming Interface (API). During this step, the device is enabled, its base address registers are mapped, and its DMA mask is set,
- coherent memory allocation; to enable DMA transfers from both FPGA to CPU and CPU to FPGA, coherent buffers must be allocated and their physical addresses must be acquired. All these operations are carried out using the `dmam_alloc_coherent` function,
- miscellaneous device registration; to enable support for `read`, `write`, and `mmap` system calls, a character device is registered using a miscellaneous device framework,
- FPGA configuration; at the end of the initialization sequence, IPs control registers are set, e.g., the physical addresses of the allocated coherent buffers are written into the DMA engine control registers.

3.1.2 Userspace interface

During a device registration, the driver creates a devfs node. This file is used as an interface between user space processes and the module. It allows applications to communicate with the driver and, as a result, interact with the FPGA. The following system calls are supported by the created node:

- `read`; used to transfer data from the FPGA OCM to the CPU DDR,
- `write`; used to transfer data from the CPU DDR to the FPGA OCM,
- `mmap`; used to translate an FPGA internal address space into a process virtual one.

3.2 Shared libraries

A common code used by both an interactive controller and a non-interactive tester is stored in shared libraries. These software components are built of C++ classes that implement low-level functionalities, such as system management and diagnostic message logging. This approach eliminates code repetitions, reduces the size of generated binaries, and enables code reuse in future projects.

3.3 Interactive system controller

The system controller is a Qt-based C++ application that enables interactive system control. It allows the user to configure the system parameters, modify a tested integrated circuit configuration, and perform the X-ray registration. This application has many features that are useful in interactive detector testing, such as acquired data exporting, chip configuration loading from JavaScript Object Notation (JSON) files, and parametrized threshold scan execution.

3.4 Non-interactive system tester

The tester is a GoogleTest-based C++ application used for automatic system testing. It allows a user to execute test suites that verify the correctness of the system. The developed system tester facilitates the definition and execution of tests, e.g. those that determine the tested IC parameters.

4 Test results

One of the tests carried out to prove the correct operation of the system was calibration of the discriminators offsets in pixels. The offsets are measured by scanning the discriminators threshold voltages without providing an input signal provided. As the output of a discriminator is connected to a counter, the maximum number of counts is registered when the threshold is equal to the offset value of the discriminator.

Each discriminator has a trimming DAC to tune the offset voltage. The discrimination threshold scanning for different trimming DAC values allows to find an optimal DAC setting.

Figures 7(a) and 7(b) present the results of discrimination threshold scans before and after offset trimming, respectively. The discriminators offsets spread is reduced from 19.33 mV to 9.58 mV (std. dev.).

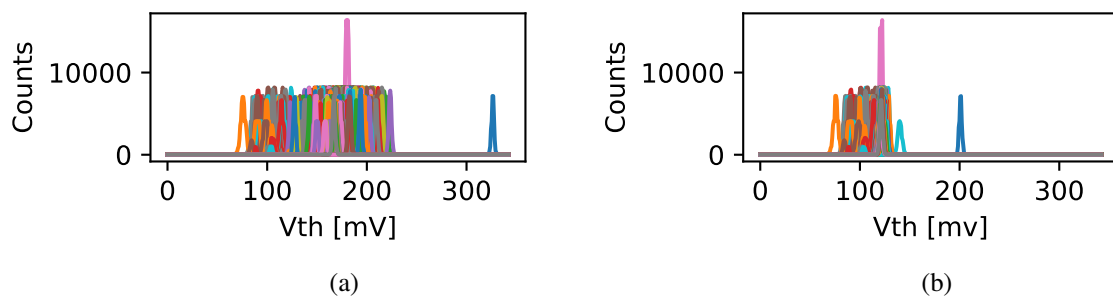


Figure 7. Noise counts registered for different threshold voltages a) before and b) after offset trimming.

5 Comparison with existing solutions

The most important tasks performed by the readout and test systems for pixelated ICs are IC control and data acquisition. There are many approaches to achieving these tasks, of which the usage of microprocessors and FPGAs is especially popular.

Because a single command may produce a huge amount of data, such as a command that triggers X-ray registration, control commands may be transmitted with a lower throughput than data transferred in the opposite direction. Although FPGAs allow to implement custom protocols to exchange data with tested ICs, some standardized solutions must be used to transfer data to PCs.

An interesting way to fulfill the mentioned requirement is the usage of Universal Serial Bus (USB) 3.0 cables allowing to construct inexpensive solutions providing throughput up to 5 Gbps [6]. Systems in which the connection between microprocessors and FPGAs is based on Ethernet links may achieve similar data transfer performance, providing a socket-based software interface [7]. The PCIe interface, which is used in the presented solution, allows significantly higher throughput (up to ~ 7.8 Gbps) but requires placing the FPGA board inside a PCIe connector of the PC motherboard. The completely different approach for IC read data analysis may be achieved through the integration of a microprocessor and pixels in the same silicon substrate [2], which allows to analyze data on-chip but still requires transferring data to a device with a larger storage space.

Each of the discussed solutions has some unique advantages but the one presented in this article is undoubtedly the most suitable for systems requiring high data throughput.

6 Conclusions

In this article, the system for fast readout and tests of the pixel IC operating in a single-photon counting mode is presented. The proposed solution is based on the Intel Arria 10 GX development kit and a Linux-powered PC connected via the PCIe 8x Gen 3 interface. The overall architecture of the system is based on standard interfaces and solutions, such as the Avalon-MM bus and PCIe interface, and may be easily adapted to control and test other pixel ICs.

The presented solution was used to test the readout IC for the pixel detector, used in X-ray imaging applications. It allowed to determine and calibrate discriminators offsets spread. The presented solution was also briefly compared in this article with some existing solutions.

One of the directions considered by the Authors for the further development of the presented solution is the enablement of the multigigabit transceivers integrated inside the FPGA. The use of these components will significantly increase the number of potential applications.

Acknowledgments

The work was partially supported by the National Center for Research and Development, Poland contract no MAZOWSZE/0099/19 for PID Polska.

References

- [1] R. Ballabriga et al., *Review of hybrid pixel detector readout ASICs for spectroscopic X-ray imaging*, 2016 *JINST* **11** P01007.
- [2] P. Skrzypiec and R. Szczygieł, *Readout chip with RISC-V microprocessor for hybrid pixel detectors*, 2023 *JINST* **18** C01030.
- [3] K. Kasinski et al., *Characterization of the STS/MUCH-XYTER2, a 128-channel time and amplitude measurement IC for gas and silicon microstrip sensors*, *Nucl. Instrum. Meth. A* **908** (2018) 225.
- [4] A. Koziol et al., *Evaluation of the UFXC32k photon-counting detector for pump-probe experiments using synchrotron radiation*, *J. Synchrotron Radiat.* **25** (2018) 413.
- [5] K. Chorągiewicz, *Integrated Pixel X-ray Photon Counter For Food Industry*, in the proceedings of the ELTE, Ryn, Poland (2023).
- [6] M. Zoladz et al., *UFXC32k based camera module with a custom soft processor and USB 3.0 for large area detectors*, 2018 *JINST* **13** P01017.
- [7] M. Daas et al., *BDAQ53, a versatile pixel detector readout and test system for the ATLAS and CMS HL-LHC upgrades*, *Nucl. Instrum. Meth. A* **986** (2021) 164721 [[arXiv:2005.11225](https://arxiv.org/abs/2005.11225)].