

Ultra-Fast Streaming Camera Platform for Scientific Applications

Michele Caselle, Suren Chilingaryan, Armin Herth, Andreas Kopmann, Uros Stevanovic, Matthias Vogelgesang, Matthias Balzer, Marc Weber

Abstract—X-ray computed tomography (CT) is a method for non-destructive investigation. Three-dimensional images of internal structure can be reconstructed using a two-dimensional detector. The poly-chromatic high density photon flux in the modern synchrotron light sources offers hard X-ray imaging with a spatio-temporal resolution up to the μm - μs range. Existing indirect X-ray image detectors can be adapted for fast image acquisition by employing CMOS-based digital high speed camera. In this paper, we propose a high-speed visible light camera based on commercial CMOS sensor with embedded processing implemented in FPGA. This platform has been used to develop a novel architecture for a self-event trigger. This feature is able to increase the original frame rate of the CMOS sensor and reduce the amount of the received data. Thanks to a low noise design, high frame rate (kilohertz range) and high speed data transfer, this camera can be employed in modern synchrotron ultra-fast X-ray radiography and computed tomography. The camera setup is accomplished by high-throughput Linux drivers and a seamless integration in our GPU computing framework. Selected applications from life sciences and materials research underline the high potential of this high-speed camera in a hard X-ray micro-imaging approach.

Keywords: high-speed imaging, CMOS-image sensor, FPGA real-time parallel data processing.

I. INTRODUCTION

SYNCHROTRON-BASED X-ray tomographic microscopy is a powerful technique for non-destructive, high-resolution investigation of broad range of samples, including live sciences. High-brilliance and high-coherence in the modern synchrotron radiation facilities allow micrometer and sub-micrometer three-dimensional imaging within a very short time. At the ANKA synchrotron radiation source [1] located at the Karlsruhe Institute of Technology (KIT) in Germany [2], a new undulator/wiggler beamline for X-ray imaging is under construction [3]. A novel concept of ultra-fast and micro-imaging X-ray computed tomography experiment station will be installed at this beamline. The project 'Ultra Fast X-ray imaging of scientific processes with On-line assessment and data-driven process control' (UFO) aims to develop the next generation of X-ray computed tomography experimental stations optimized for 3D and 4D X-ray imaging. The additional resolution in the time domain gives insight in the temporal structure evolution allowing to understand functionality of devices and organisms and to optimize technological processes. The architecture of UFO is presented in Fig. 1. The whole setup consists of three sections, the beamline, that supplies

high flux density to the sample position, the UFO experimental station, and a high-performance data storage system.

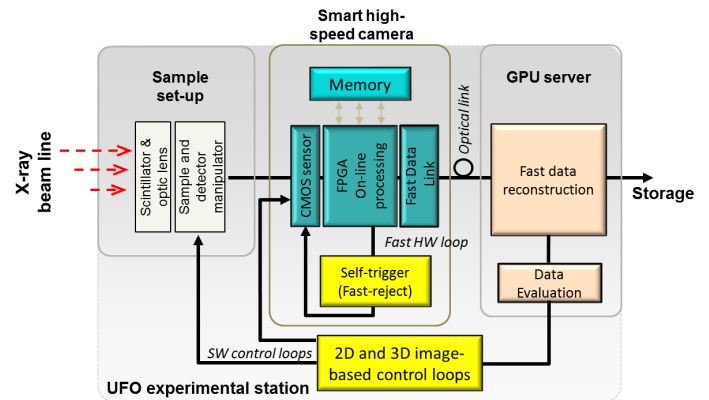


Fig. 1. UFO experiment station and feedback loops process control

The UFO experimental station, shown in Fig. 1, consists of the following main functional units:

- *Sample setup.* A dedicated setup for fast sample manipulation (tomographic and laminographic rotations and scanning translations) and adequate sample loading.
- *Smart high-speed camera.* A fully programmable and high performance smart camera that uses a local hardware feedback loop as a self-trigger.
- *GPU server and on-line data processing and evaluation.* Graphics processors for accelerating the 3D image reconstruction. The gained speed-up, for the first time, will enable an on-line feedback loop for sample manipulations and automatic adjustment of the optical system.

UFO is based on the indirect detection method where a crystal converter screen (scintillator) is optically coupled to a CMOS-based digital camera. In this approach the pixel detector is not placed along the X-ray beamline and consequently a rad-tolerant silicon pixel detector is not needed. This opens several possibilities in the field of the commercial CCD or CMOS sensor devices. The requirements concerning the high-speed data transfer, the image-based process control and the fully programmable camera for an automatic adaptation to the experiment conditions are only partially met by available commercial cameras. For these reasons, Institute for Data Processing and Electronics (IPE) at KIT has started to develop the first prototype of a made-in-house camera that fulfills the UFO requirements.

II. HIGH SPEED CAMERA ARCHITECTURE

Advances in CMOS image-sensor technology are giving rise to a new generation of high-speed cameras to capture events previously impossible to be acquired by conventional CCD cameras [4].

The present FPGA devices offer a large number of high-speed I/O interconnects combined with a large number of native blocks like DSP, FIFO and RAM, PLL and others. For these reasons the FPGAs are usually employed for the control and real-time data processing of the high-data throughput CMOS-image sensor.

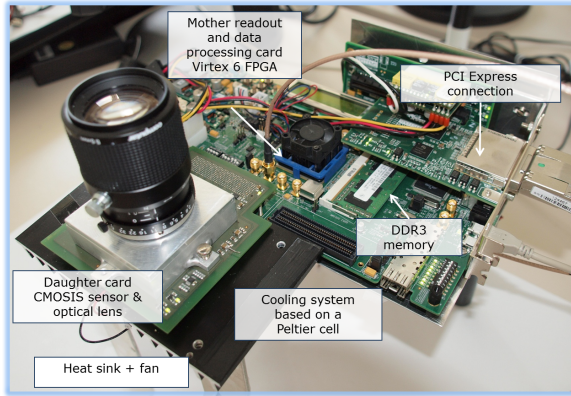


Fig. 2. First prototype of so-called 'UFO-camera'

This concept has been employed in the design of the smart high-speed camera that will be installed in the UFO experiment station. The first camera prototype is based on a commercial CMOS 2.2 Mpixels image-sensor [5] with a moderate frame rate of 340 fps. The camera system consists of a daughter acquisition card where only the CMOS-image sensor is mounted and an FPGA-based mother card for readout and data processing. The camera is shown in Fig. 2. The daughter and mother cards are connected by high speed and high density FMC-Samtec connector [6].

Typically, CMOS-image sensors used in scientific applications are cooled to reduce dark current generation. Therefore a cooling system based on a Peltier cell is employed. The Peltier cell is also controlled by the FPGA.

A functional diagram of the camera prototype is shown in Fig. 3. The readout architecture can be divided into three main parts: CMOS-image sensor, mother-board based on the Virtex6 Xilinx FPGA [7] and the PC used for camera control and Data Acquisition System (DAQ).

The CMOS-image sensor receives a request for the new frame in combination with the defined exposure time. When the integration time is finished, the image stored in the pixel-matrix (global shutter) is read out sequentially, row-by-row. The pixel values are then passed to a column ADC cell, in which ADC conversion is performed. The digital signals are then read out over 16 parallels LVDS (low voltage differential signal) channels where each LVDS channel reads out 128 adjacent columns of the array. Control registers are foreseen for the programming of the sensor.

In order to read out the CMOS sensor as fast as possible, a PCI Express interface is used to transfer the data arriving

from the mother board directly to the computer embedded memory. A PCI Express x4 lane generation 2 standard, with a theoretical bandwidth of 20 Gbit/s, is used. The achieved bandwidth is only limited by the speed grade of the FPGA. In order to fully benefit from the high bandwidth of the PCI Express link, we use Direct Memory Access (DMA) to transfer the data from the PCI Express interface to the central system memory and vice versa. By using PC memory, the camera benefits from the evolution of memory size and frequency.

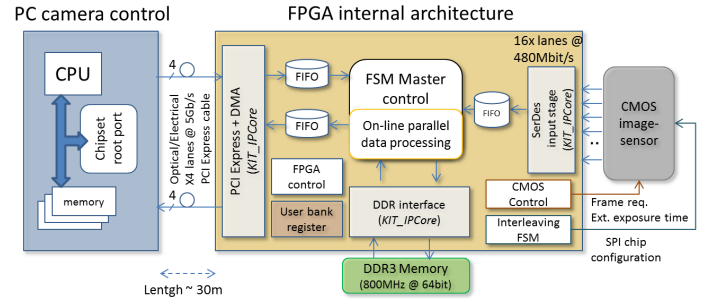


Fig. 3. Readout chain and FPGA architecture

Addressable 32-bit user bank registers are implemented in the dedicated Base Address Registers (BAR) space. Bank registers are used to write/read the status/configurations of the DMA engines, the CMOSIS chip and the FPGA logic. Unused address locations of the bank can be easily extended to any user application. The DDR (Double Data Rate) memory device is used for both temporary frame data storage before the transferring of the data and for an on-line data elaboration. The main features, implemented and tested, include:

- *Fully configurable camera.* Full access of the pixel parameters in order to adapt the pixel response at any experiment condition, like adjustable image exposure time and pixel dynamic range, noise threshold, mask, analog gain, etc. (see section V).
- *Full streaming data acquisition architecture.* Continuous data acquisition at full speed for each frame rate conditions without any readout dead time.
- *On-line image-based self-event trigger architecture (Fast reject).* Fast in-camera event recognition capable to reject frames or part of frames that do not contain new valuable information.
- *Region-of-Interest readout strategy using self-event trigger information.* Intelligent selection of the frame region-of-interest for reducing the amount of output data and, at the same time, to significantly increase the camera frame rate.
- *Easily extendable to any available CMOS-image sensor* Flexible FPGA architecture is easily adapted to any CMOS-image sensor devices.

The readout chain, shown in Fig. 3, including the driver layer, is tested and characterized. Several millions of the data packets generated by the PC have been sent to the FPGA-board. Data received from the DMA are stored in the DDR memory. At the same time a readout request from PC is received from the board and the data previously stored in DDR are sent back to the PC by the DMA engine. The data are

compared in the PC for data consistency and bit error rate estimations. Fig. 4 shows the performance of the data transfer between both PC to DDR and DDR to PC as a function of the packet size. The configuration of the PC used in this test has a

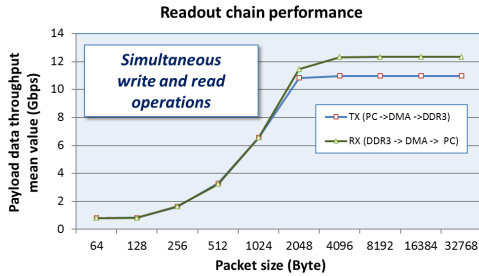


Fig. 4. Readout chain and FPGA architecture

significant impact on these measurements. The data bandwidth depicted in Fig. 4 has been estimated using a standard PC Intel dual-core with 2 GByte of DDR3 system memory and PCI Express GEN2. When the GPU-UFO infrastructure, shown in Fig. 1, is used to transfer the data between the camera and UFO-server the bandwidth increases to 16 Gbit/s in both directions. In any case, the bandwidth of 16 Gbit/s is only limited by the current FPGA speed-grade.

The camera has achieved 340 frames/s (with short exposure time) at the full resolution of 2.2 Mpixel and several thousand frames/s with the on-line self-event trigger or with the reduced/interpolated resolution. Frame rate and resolution are only limited by the current sensor data throughput of 7.7 Gbit/s. The readout architecture itself is usable for other sensors with higher frame rates and higher resolutions. The on-line self-trigger data processing implemented in the FPGA is able to handle up to several 10^9 pixels per second.

As shown in Fig. 3, three Intellectual Property (IP) logic blocks have been developed at KIT and integrated in the UFO framework in order to improve the FPGA modularity and to overcome the limitation presented in the native FPGA IPCore. In the next subsection the architecture and the performance of these three 'KIT-IPCore' logic will be discussed.

A. Fast SerDes input stage

SerDes (serializers/deserializers) are devices that can take wide bit-width, single-ended signal buses and compress them to a few, typically one, differential signal that works at a much higher frequency rate than a wide single-ended data bus. SerDes enable point-to-point movement of a large amount of data.

The CMOSIS-image sensor employed in the camera prototype uses 16 parallel high-frequency LVDS serial lines to move data from the pixel-matrix to the receiver device. Each line works at a double data rate with 480Mbits/s. Moreover, 10-bit or 12-bit ADC pixel data can be selected by the user. This has an impact in the SerDes logic that must switch the serial input to 10-bit or 12-bit parallel output. To overcome the native SerDes logic limitations present in the Virtex6 FPGA [8] a new SerDes input stage module has been developed and employed in the current design. The basic architecture of a

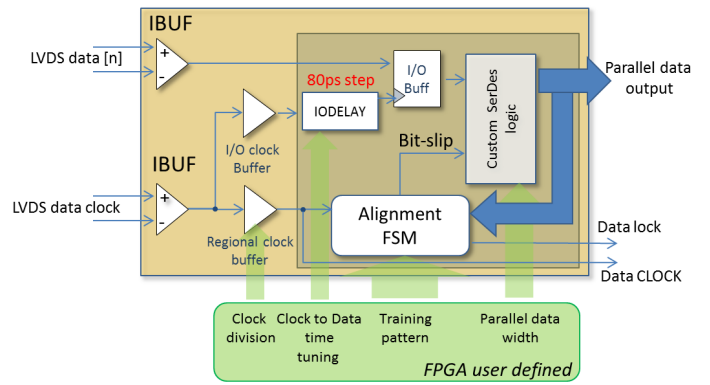


Fig. 5. SerDes block logic architecture

single SerDes channel is shown in Fig. 5. In order to cover all CMOSIS outputs, 16 parallel SerDes input stages, one for each CMOSIS output, are employed. A common FPGA regional clock [9] for all 16 input stages has been defined as a division of the LVDS data clock in according to the parallel data width. An individual programmable absolute delay primitive block, IODELAY [8], can be used for a precise 80ps step time synchronization between data-to-clock. The LVDS input data line is converted from double-data-rate to two single-ended data lines by a double-data-rate (IDDR) register [8]. Lines are then combined for a parallel data output by the custom SerDes logic. The parallel data output is compared with a training pattern for checking the correct position of the MSB by the dedicated word alignment Finite State Machine (FSM). A bit-slip signal is generated from the alignment FSM and received from the custom SerDes in order to shift the wrong MSB bit to the correct position. A data lock signal is set to inform the rest of the logic that the parallel data are correctly aligned. The presented logic block can be used for a parallel data output with a width of up to 16 bit. The clock division, parallel data width and the training pattern are FPGA reconfigurable in according to the CMOS-image sensor specifications.

B. PCI Express-DMA architecture

The term Bus Master, used in the context of PCI Express, indicates the ability of a PCIe port to initiate PCIe transactions, typically Memory Read and Write transactions. The most common application for Bus Mastering Endpoints is for DMA. DMA is a technique used for efficient transfer of data to and from host CPU system memory. This implementation presents many advantages over standard programmed input/output (PIO) data transfers. In addition, the DMA engine offloads the CPU from directly transferring the data, resulting in better overall system performance through lower CPU utilization. The KIT PCI Express-DMA architecture is based on the bus master DMA implementation developed in order to minimize the control and status signals required to handle complex PCI Express-DMA architecture. The block diagram is shown in Fig. 6. Two IPCores are employed in combination with the logic blocks developed at KIT. An integrated Endpoint Xilinx-IPcore for PCI Express [10] and two DMA [11] engines are used to move the data from the FPGA board to the PC central

memory and vice versa.

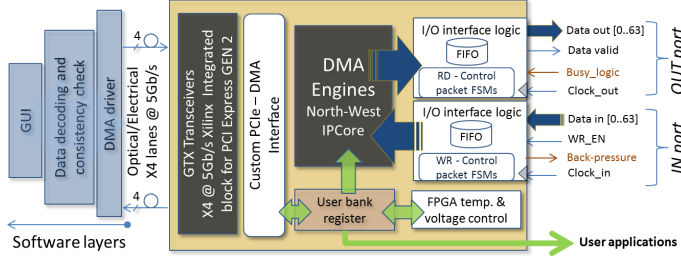


Fig. 6. PCI Express - DMA architecture

A custom PCIe-DMA interface logic has been developed to adapt the native PCI Express interface to the DMA engines. Easy signal handshaking with the internal FPGA logic is provided by two I/O interface logic blocks. Each logic block includes one data FIFO and FSMs developed to control the coherence between the received and transmitted data packets. The FSMs generate the start and end of packet signals for DMA and manage the busy and the back-pressure signals. The FIFOs are used as a temporary data accumulator and for frequency domain change. In this way the user defined clocks, *Clock_in* and *Clock_out* in Fig. 6, can be used in order to send/receive data. The *Data_out* and *Data_valid* signals will be synchronized with the user defined *Clock_out* domain. The *Data_valid* signal is used to inform the logic when the valid data are present on the data out bus. A busy signal could be used in order to temporary interrupt data flow received from the FPGA internal logic. By the *Data_in* bus a data word with a user defined clock frequency *Clock_in* is written in the logic block by a *WR_EN* signal. The *Back_pressure* signal is used to inform the logic that the driver-PC is in the busy status. A software driver layer, fully compatible with this architecture, has been developed in 64 bit@Linux operating system and will be presented in section III.

C. DDR memory interface logic

The memory interface solutions developed at KIT combine a native Xilinx physical layer (PHY) for DDR3 devices with additional block logic developed in order to extend the memory interface Xilinx IPCore [12] features and overcome some limitations present in the native DDR3 Xilinx IPCore.

By the *Data_in* bus a data word with a user defined data width (N) is written in the logic block by a *WR_EN* signal. A WR-FIFO is used as a temporary data storage and for frequency domain change between a user clock frequency, *Clock_in* in Fig. 7, to the internal logic clock domain. The Arbiter FSM continuously check. If the WR-FIFO is empty, if not, the enable signal for the write operation is propagated to the WR-DDR FSM. The WR-DDR FSM receives the write command and generates the address and all control signals for the PHY logic. The PHY logic receives the command and writes the new data on the address position specified by the WR-DDR FSM.

The Arbiter FSM can receive a read request for the DDR3 device. In this case, the Arbiter check if the WR-DDR FSM is in 'idle' state. If yes, a read command is propagated to the

RD-DDR FSM. The RD-DDR FSM receives a read command and generates the address and all control signals for the PHY logic. The output data from the PHY are stored in the RD-FIFO. In according to the user defined data width (M), a data word is present at the *Data_out* port. A *Data_valid* signal is provided to inform the user logic that the data are ready. All signals are synchronized to the user defined *Clock_out*.

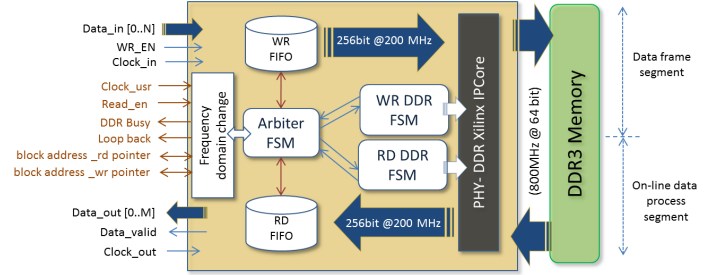


Fig. 7. DDR3 memory device interface

As shown in Fig. 7 the internal write and read paths work with a data width of 256 bits at 200 MHz internal clock that corresponds to the bandwidth of 51 Gbit/s in a half-duplex mode. This bandwidth limitation is imposed from the PHY logic in the present FPGA speed-grade.

A quasi full-duplex data flow is achieved with the proposed architecture. Balance optimization between the amount of the data in both FIFOs is managed by the Arbiter FSM. Thanks to this balance, the intelligent burst write and read commands can be propagated to the PHY in an alternating way. This is equivalent at the full-duplex DDR3 interface with a mean bandwidth of 25 Gbit/s in each direction.

III. ADVANCED LINUX PCI SERVICES (ALPS) FOR RAPID PROTOTYPING OF PCI-BASED DAQ ELECTRONICS

We developed a universal PCI driver and a debugging tool to support hardware development of PCI/PCI Express DAQ electronics like the presented camera prototype.

The basic ideas behind ALPS are:

- *Development phase.* During the development phase for the majority of projects only a standard set of functionality is required. To prevent blocking hardware development by missing or malfunctioning software, the required functionality can be provided by a universal PCI driver.
- *Implementation ability.* If necessary, a dedicated driver may be implemented when the hardware is ready. The driver may be based on the universal driver and extend its functionality on the level of driver or user-space software. Our design is optimized for high data throughput and provides place-holders to simplify extension by the device-specific functions.
- *Compatibility.* Adjustments for new versions of Linux kernel are often required. Therefore, the kernel module is kept as small as possible. It is only responsible for device configuration, interrupt counting, management of DMA buffers, and mapping of both PCI I/O space and DMA buffers into the userspace. All other functions including

the actual implementation of DMA engine are realized in user space.

- *Fine grained scripting.* An example would be to start the DMA engine, set some registers to initiate DMA transfer, read data from DMA engine, make an attempt to process it, and if the wrong data is returned, analyze the status registers to determine the occurred problem. No software modifications are required during hardware debugging.
- *Integration.* To simplify integration with 3rd party components and into the distributed DAQ systems, web service interface for all SDK functions will be provided.

The architecture of the ALPS is given in figure Fig. 8. It

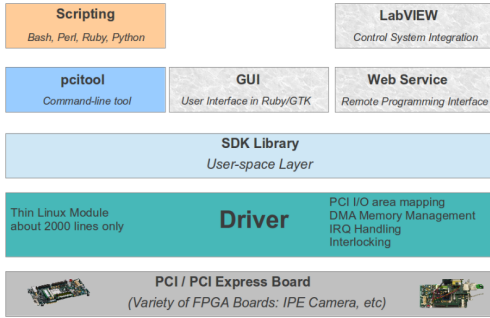


Fig. 8. ALPS architecture

includes a kernel module, an SDK library, and a command-line utility. To simplify integration with distributed data acquisition systems, an enhancement of ALPS by a web-service interface is planned.

The Fig. 9 illustrates architecture of SDK library. The main components are:

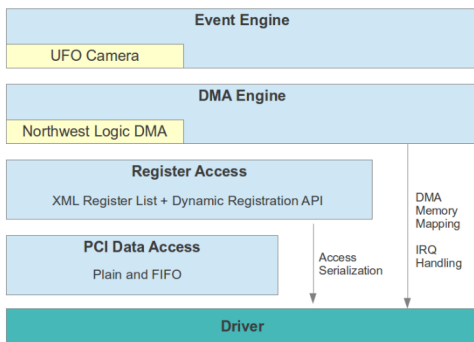


Fig. 9. Structure of the SDK library

- *PCI Data Access layer:* provides raw access to PCI standard I/O memory.
- *Register Access layer:* defines a register model. The registers are defined by an XML file. Additional registers may be added at run-time (also used by DMA engine). Integer registers with sizes up to 64 bits are supported. The register size and alignment may be specified with bit precision. The endianness conversions is handled automatically if the register endianness is specified. Custom functions to read/write registers may be provided by plugins.

- *DMA Engine layer:* is a pluggable interface for DMA engines. The basic interface defines 5 methods: start/stop DMA channel, read/write data from/to DMA channel and stream data from DMA channel to supplied callback function.
- *Event Engine layer:* defines an event-based model to integrate device-specific code (by plugins). Each device can define multiple events and for each event several data types. The events will be triggered in hardware or requested by software. The client application may subscribe to get event notifications. Upon event notification, the application can request the desired type of data.

The universal driver is successfully used for the development of the first prototype of the high-throughput camera. The camera-specific functions are implemented using the Event Engine.

IV. SELF EVENT TRIGGER (FAST FRAME REJECT) AND HIGH FRAME RATE READOUT STRATEGIES

A. Reason for a fast event trigger mechanism

Fast processes which cannot be controlled by an external signal require data recording at high frame rates. Unpredictable physical events could be lost or partially acquired due to limited observation time given by the camera memory and/or readout bandwidth limitations. An intelligent image-based self-trigger for these kind of applications have been developed for the presented camera platform. An adaptive frame rate and adaptive frame size of the event is used in order to record the temporal evolution of the physical events.

An example of occurrence of fast events is shown in Fig. 10, where two bubbles in gelatinous agar merge. The gelatinous bubbles are unchanged for the first 48.9 ms of the data taking, corresponding to 489 redundant frames at 10 kframe/s. After that, physical events happen in a very short time.

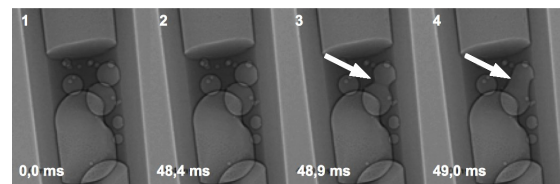


Fig. 10. Bubble images in X-ray material science radiography

The developed logic is able to detect fast physical events and consequently, reject redundant data frames. A multi-event detection located in different frame regions is possible. Additional benefits are: simplification and acceleration of data analysis, optimization of the effective bandwidth and significant increase of the frame rate.

B. Self-event trigger algorithm and FPGA implementation

Recent CMOS sensors support direct pixel access. This enables readout of individual rows of the pixel-matrix supports readout strategies to increasing the frame rate and/or reducing the amount of data. The fast reject logic uses a row-based sub-sampling interleaving readout mechanism. This allows a

drastic reduction of the readout time without losing the full field of view. In interleaved mode several rows are omitted between the selected rows. The number of skipped rows is programmable by bank register and defines the maximal possible frame rate.

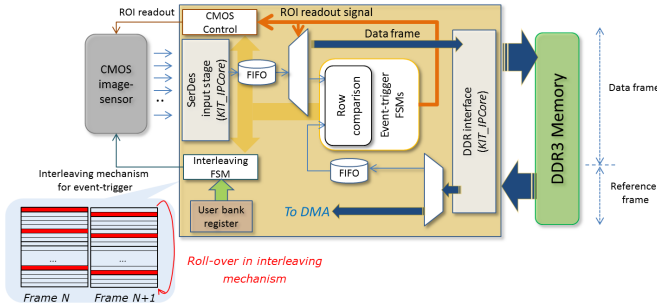


Fig. 11. 'Self-event trigger (Fast Reject) and the ROI strategy'

The self-event trigger and region-of-interest readout architecture is shown in Fig. 11. The reference frame (complete frame) is stored in the dedicated DDR3 memory segment at the beginning of the data taking and then the event-trigger FSM will enable the interleaving frame mechanism. In each interleaving frame the first sent row is shifted down by one row position in order to have a roll-over readout mechanism as shown in Fig. 11. Corresponding rows of the current interleaved frame and the previously stored reference frame are compared and checked for differences. In case of a significant difference, the row is 'token' as a candidate for a trigger. The event-trigger FSMs receive the 'token' rows and uses these signals for a dynamical programming of the region-of-interest that will be acquired.

In order to balance the efficiency of the event detection and the noise influence, several programmable thresholds are used. *Pixel threshold*: used to reduce the noise in the pixel count value. *Row threshold*: indicates how many positively compared pixels are needed to 'token' the row. *Global threshold*: how many 'token' rows are needed in order to generate the trigger signal.

The **intelligent Region-of-Interest readout** combined with the self-event trigger (Fast Reject) represents the best compromise between the high-spatial resolution and time resolution and is advised for fast X-ray micro-imaging radiography. The additional benefit of this method is maintaining the full point of view of the scene. In Fig. 12 the frame rate achieved by the logic described as a function of the number of skipped rows is shown. The frame rate has been evaluated considering both the reaction time, to detect the physical event, and the intelligent region-of-interest readout time. The results have been evaluated for two different physical event areas. The reaction time of the self-event trigger logic depends on the number of skipped rows and the physical event area. It is understandable that the larger area of changes are faster to detect with the larger gap between rows and, therefore, we expect a fast reaction time (less than 100us), in this condition the readout time limited the frame rate. From other side, the reaction time became dominant when the fraction of area needed to detect is smaller than the gap between rows, in

this condition several interleaving frames are needed to detect the physical event. This method increases the original CMOS-image sensor frame rate up to a factor of 10.

A special test station setup based on a pulse laser beam, mounted on the X-Y motorized axes with micrometer precision, is under preparation for an exhaustive characterization of this readout strategy.

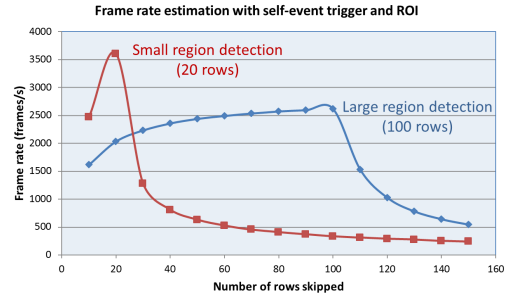


Fig. 12. Interleaving readout implemented in FPGA for high frame rate

Two additional kilo frames rate readout modes are possible with the proposed architecture. The first method consists of a **windowing readout** strategy. A portion (window) of the pixel matrix can be read with a fast frame rate by the CMOS control module as shown in Fig. 11. This method keeps a high spatial and temporal resolution but with a reduced field of view. The number of lines and the first line can be set in the appropriate bank register. Up to 8 windows can be defined in the same frame. The achieved frame rate depends on the total area of the windows.

Unlike the windowing readout strategy, the **fully programmable interleaving readout** maintains the same field of view but with a reduced pixels resolution. The logic reads the rows in an interleaving mode as describe before and shown in Fig. 12. The missing rows in each frame are reconstructed by a spatial-temporal interpolation algorithm in the GPU-server [13]. Different FPGA configurations can be programmed by setting the appropriate registers in the bank register and as a consequence, several frame rate speeds are possible to achieve.

V. FULLY PROGRAMMABLE CAMERA - ADAPTATION AT EXPERIMENT CONDITIONS

The limited density of the photon flux in the synchrotron light source application sets the fundamental limit on image sensor performance in the high frame rate acquisition (short integration time). The temporal noise components are dominant in these conditions. For these reasons, a low noise camera is needed in order to keep a reasonable Signal-to-Noise Ratio (SNR) in these low illumination conditions. Contrary to that, a high frame resolution requires a longer integration time, and the temporal noise combined with the (Photo-response Non-Uniformity), also known as Fixed Pattern Noise (FPN), becomes dominant. In order to overcome the limitation outlined before and, also, to improve the current CMOS-image sensor advantages as well, the camera has been characterized in different camera working-points.

The camera calibrations were carried out at the Institut für Experimentelle Kernphysik (IEKP) at KIT using the emission

of characteristic "secondary" (or fluorescent) γ -energy by different metals (Al: 1.5KeV, Fe: 6.4 keV, Mo: 17.5 keV, Sn: 25.3 keV) initiated by a tungsten X-ray tube [14]. The camera calibration setup is shown in Fig. 13. This method is used to study the camera response and to obtain an absolute energy calibration.

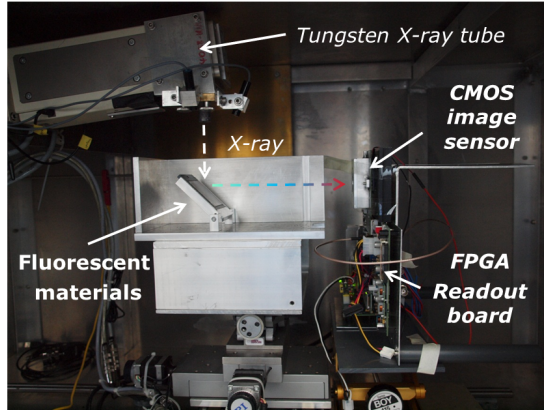


Fig. 13. IEKP X-ray calibration setup

Each fluorescent material creates a different amount of electron/hole pairs in the silicon pixel detector, the number of electron/hole pairs produced are reported in TABLE I. For each material several thousand of frames have been acquired with a proper integration time for off-line data analysis.

TABLE I
TRANSITION ELEMENT FOR X-RAY CAMERA CALIBRATION

Element	Theory k_{L3} Energy (eV)	e^-/h^+ pairs produced
Al	1487	413
Fe	6403	1779
Mo	17479	4885
Sn	25271	7019

The response of a single photon is well confined within few pixels (clusters) as the result of the signal charge sharing among adjacent pixels. For each frame all clusters were reconstructed by a clustering algorithm and the amount of the energy released in each cluster has been estimated. Correlation between photon energy (number of e/h pairs produced) and the cluster signal (in ADC counts) is shown in Fig. 14. Excellent linearity of the response of the CMOS-sensor at the different fluorescent metals was found. A precise conversion factor between electrons and ADC counts of $21 e^-/LSB$ was calculated.

As described before, with the photoelectric effect, electrons generated by the noise cannot be distinguished from the electrons generated by photons (signal). Therefore, there is always some number of electrons stored in the pixels that are not the result of photons hitting the detector. The generated noise electrons exists even when light does not hit the detector surface, and is referred to as the dark current. Dark current is a multiplicative form of noise, the level of which is proportional to the length of the integration time and sensor temperature. The total dark noise contribution has been estimated for a short

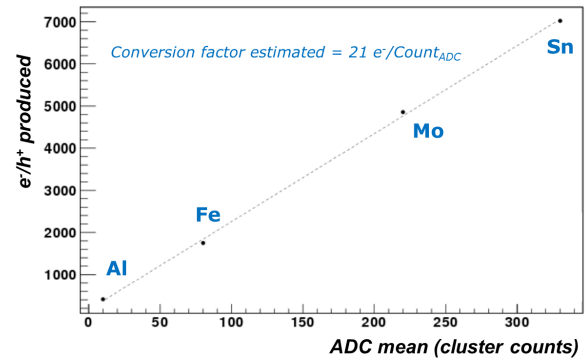


Fig. 14. The camera efficiency calibration

(high frame rate) and long (high frames resolution) integration time. The Fig. 15 reports the camera behavior of the total dark

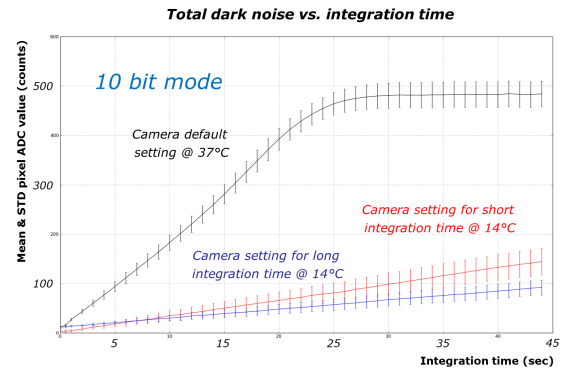


Fig. 15. The total dark noise vs. integration time at different camera settings

noise as a function of the integration time and temperature on the sensor. The influence of the temperature on the total dark noise is shown in the default camera setting (black line). In these conditions, the dark noise increases rapidly as consequence of a major thermal charge generation in the sensor elements, the saturation point of the pixel is achieved after about 25s. Two optimum settings have been found to improve the noise conditions. The first setting (blue line) improves the camera performance for longer integration time (more that 7 seconds). The second setting (red line) improves the noise for the high frame rate radiography. A total noise contribution of $87.15 e^-/s @ 14^\circ C$ was calculated using the camera for short integration time. This value is in agreement with the dark noise value of $125e^-/s @ 25^\circ C$ reported in the CMOSIS datasheet. The behavior of the standard deviation (STD) as a function of the integration time, shown in all camera settings of the Fig. 15, depends on the fixed pattern noise which increases with the signal level on the pixels. A drastic reduction of this systematic pixel-to-pixel non-uniformity contribution has been achieved with the present FPGA architecture. The architecture shown in Fig. 11 can be re-configured for a subtraction between corresponding pixels of the current data frame and a reference background frame previously stored without any target. The FPN noise correction logic can operate in full streaming mode.

The benefit of a better signal-to-noise ratio using the

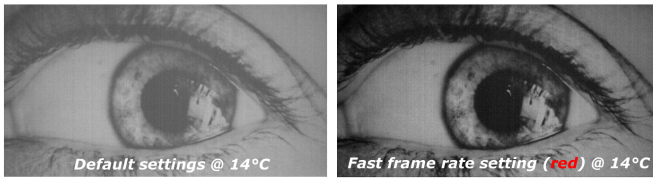


Fig. 16. Camera enhancement settings

advanced camera setting for a fast hard X-ray computed tomography is shown in figure Fig. 16. Two frames have been acquired with a lower illumination level and with 100 microsecond integration time. The first picture is taken with default camera parameters at a temperature of 14°C and exhibits a similar behavior as a commercial CCD. The second was acquired using the advanced setting 2. The comparison shows that the SNR is better in the second picture as a consequence of the reduced dark noise contribution.



Fig. 17. ANKA X-ray camera test beam

The camera prototype was integrated into the demonstrator setup at the TopoTomo beam line [15] and was successfully tested at ANKA with a moderate X-ray flux density. Several thousand of radiographies have been acquired at full speed (300 frames/s) in streaming mode, one X-ray radiography is shown in Fig. 17. The FPN noise contribution has been drastically reduced by the background subtraction, the dark noise reduced as well, an appropriate SNR level has been achieved. A spatial resolution of a few micrometers has been estimated by a dedicated sub-micrometer pattern for X-ray. The exhaustive camera characterization including the full-well sensor charge and the signal-to-noise estimation as a function of the signal level, integration time and camera setting will be executed in the new ANKA detector laboratory.

VI. CONCLUSION AND FUTURE WORK

In this paper we presented a high-throughput imaging platform for fully programmable scientific cameras. The first camera demonstrator achieves the maximum frame rate of the image sensor, with 340 fps and 2.2 Mpixel @ 10 bits and a data rate of up to 1 GB/sec. Several thousand frames/s are possible at a reduced resolution or using the intelligent self-event trigger (fast reject). Frame rate and resolution are only limited by the current sensor data throughput. An available Linux driver links the camera to our GPU compute servers.

The next generation of visible light camera is under development and will employ a faster CMOS sensor with a

readout bandwidth of 50 Gb/s and a original frame rate of 5000 frames/s. A novel made-in-house readout board will be used for the high-speed streaming ability. A higher frame rate in the range of several tens of kilo frame rate should be possible with the self-event trigger strategy described in this paper. To achieve a real-time elaboration at full speed, using a cluster of GPU-processing nodes, a fast data transfer network link, based on InfiniBand technology, will be provided.

ACKNOWLEDGMENT

The authors would like to thank to the CMOSIS and the NORTHWEST companies for the cooperation in this work, and T. Weleiss and S. Heindl of the Institut für Experimentelle Kernphysik at KIT for X-ray characterization facility.

REFERENCES

- [1] http://ankaweb.fzk.de/website.php?page=about_main
- [2] <http://www.kit.edu/index.php>
- [3] <http://www.anka-cos.kit.edu/99.php>
- [4] http://www.pco.de/fileadmin/user_upload/db/download/OLE_June_High_Speed_Camera_PG.pdf
- [5] http://www.cmosis.com/products/standard_products/cm2000
- [6] ANSI/VITA 57.1 FPGA Mezzanine Card (FMC) Standard. URL: <http://www.vita.com>
- [7] <http://www.xilinx.com/products/silicon-devices/fpga/virtex-6/index.htm>
- [8] SelectIO Resources User Guide. UG361 (v1.3) August 16, 2010.
- [9] Virtex-6 FPGA Clocking Resources (User Guide) UG362 (v1.6) January 17, 2011
- [10] Virtex-6 FPGA Integrated Block for PCI Express, UG671 (v1.0) October 5, 2010
- [11] <http://www.nwlogic.com/>
- [12] Virtex-6 FPGA Memory Interface Solutions. UG406 March 1, 2011
- [13] Chilingaryan, S. A GPU-based Architecture for Real-Time Data Assessment at Synchrotron Experiment. Supercomputing 2011, Seattle, WA 12-18.11.2011.
- [14] M. Guthoff. Geant4 simulation of a filtered X-ray source for radiation damage studies. Nuclear Instruments and Methods in Physics Research A 675 (2012)
- [15] A. Rack. The micro-imaging station of the TopoTomo beamline at the ANKA synchrotron light source. NIMB-S-08-00848, R3