

Hardware/Software Co-Design of a control and data acquisition system for Computed Tomography

Daniele Passaretti^{1,2}, Thilo Pionteck¹

¹Otto von Guericke University Magdeburg, ²Research Campus STIMULATE, Magdeburg, Germany
daniele.passaretti@ovgu.de, thilo.pionteck@ovgu.de

Abstract—In the last decades, the continuous advances of computed tomography sensor technology and the different medical use cases have opened new challenges for data acquisition systems: i.e. controlling and synchronising different sub-systems, and acquiring and managing the huge amount of data, under real-time requirements. System-on-Chip-FPGAs are perfectly suited for all these tasks that involve hardware and software. In this paper, we propose a new acquisition system architecture for Clinical Computed Tomography (CT) for SoC, based on Hardware/Software Co-Design methodology. We discuss how to partition the different tasks (controlling, acquisition, synchronisation) on Programmable Logic (PL) and Processing System (PS) parts. Moreover, we analyse the interactions between these parts for having the performance required from the CT application. This architecture is able to control and synchronise all components inside the CT, acquire data via the optical channel, and process real-time image data in the order of 10 Gb/s, inside a single SoC.

Index Terms—Computed tomography, accelerator architectures, high-performance communication, Hardware/Software Co-Design, System-on-Chip.

I. INTRODUCTION

Computed Tomography (CT) is one of the most common medical radiology techniques; it is a non-invasive procedure for 2D/3D images of the patients' internal organs [1]. With the introduction of the Multi-Modality and Interventional Radiology, emerges the need for a system in which the doctor can interact with the CT. Thus, it must provide a real-time image stream during the surgery. These medical applications are strictly supported by technological enhancements that involve new hardware/software architectures that are able to acquire data with high throughput. Furthermore, they must process image data and coordinate the different modules inside CT for having less X-ray dose and showing real-time images.

Due to time-to-market, the small market of CT and the continuous advances of the sensor technology, the hardware controlling part of the CT is mainly composed of FPGAs. Indeed, Xilinx and Altera proposed different architectural models for Clinical CT devices [7], [5]; They did an overview of the FPGA-SoC architectures inside CT devices, based on the different CT tasks. Moreover, different companies produce acquisition systems for the camera and detector data. They are all FPGA-based and can be customized for different applications and speed, and they can be reprogrammable for different detector sub-systems. In this paper, we propose a new Hardware/Software Architecture for controlling the devices inside CT and for managing the data acquisition flow in the

system. Based on the different partitioned tasks, we focus on the hardware/software co-design of our architecture on System-on-Chip (SoC) with FPGA inside.

Before presenting our hardware/software co-design architecture for SoC-FPGAs, we introduce the reader to the Computed Tomography appliance and problem, in Sec. II. After that, in the Sec. III we introduce the related works from Xilinx and Altera and other acquisition systems in the literature. In Sec. IV, we introduce the main classified tasks and how they match in software and hardware in the proposed architecture. Finally, in Sec. V we describe the implementation of our architecture on FPGA and its integration on real CT appliance, with the related evaluation in terms of area and throughput.

II. BACKGROUND: COMPUTED TOMOGRAPHY

In this section, we describe the Computed Tomography appliance. It is composed of: a gantry module, detectors' array system (DMS), an X-ray tube system, collimators, a patient's table and an image reconstruction unit, as shown in Fig. 1. The gantry module, patient's table and an image reconstruction unit are fixed on the ground, called stationary side; all the other components are fixed on the rotating disk of the gantry, called rotating side [1] [2]. They communicate between each other via the slip-ring technology [3], [6]. The disk on the gantry rotates typically with a frequency of ~ 170 rpm, and the detector can produce thousands of projections (shots) of the patient per round [8]. The data must arrive from the detector to the reconstruction system via slip-ring technology that is often the bottleneck of the system [3], [6].

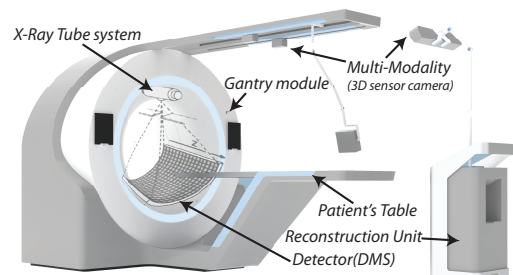


Fig. 1. Computed Tomography appliance, [2]

Due to this mechanical physical issue, one or more control and data acquisition systems are required on the rotating and stationary sides. These are responsible for acquiring,

compressing, and managing the data, and for controlling and synchronising the different modules in the system. For example, the gantry position, DMS position and the X-ray tube current and voltage are synchronised with the patient's table, so the parameters need to be updated in real-time.

All these components are internally designed with custom architectures, protocols and software. Because of this, they must be controlled and synchronised from a *custom acquisition system*, often called *frame-grabber*. It is able to manage different tasks and hardware/software protocols. Moreover, it must fuse control and data-flow in the CT scenarios. In Sec. IV we present our *custom acquisition system* based on Hardware/Software Co-Design integration and implemented on SoC FPGA.

III. RELATED WORKS

In literature, different companies (i.e. Xilinx, Intel-Altera, Eltec) have proposed their *frame-grabber* architectures for camera and CT device. Altera-Intel [7] proposes an architectural model based on three modules: Data Acquisition Card, Data Consolidation Card and Data Processing Card. Usually, as described by Altera in [7], there are more than one *Data Acquisition Card* inside the DMS, because of thousands of detection elements (DEs), e.g., 43008 DEs in Philips CD300 [8]. For this layer, they suggest using FPGAs with a huge number of A/D converters and DSPs for filtering (Cyclone/Arria family). The data from DMS (multi-lane) are transmitted on *Data Consolidation Card* that is responsible for buffering, aligning and sending data to the *Data Processing Card* through the slip-ring technology (single-lane). In this layer, they use a SoC-FPGA device (Stratix family); it allows managing and combining various control tasks and data transmissions. For reconstruction, they put an FPGA in the *Data Processing Card*. This have the following features: many DSPs, BRAM and DRAM memory, which can reach over 10,000 GFLOPS [9]. This yields comparable performance to modern GPUs.

Xilinx [5] divides the CT device in 4 sub-systems: *HV Supply Control*, *Data Acquisition & Gantry Control*, *Image Reconstruction* and *System Sequencer*. The *HV Supply Control* manages all the tasks inside the X-ray tube: software parameters, high voltage power supply and controlling and data acquisition errors. The *Data Acquisition & Gantry Control* manages the DMS and forwards the data to the *Image Reconstruction*. The *Image Reconstruction* and *System Sequencer* implement the reconstruction algorithm and the synchronization tasks inside the CT. For the reconstruction algorithm, Xilinx proposes Versal ACAP [5], a new heterogeneous architecture with FPGA, CPU and engine architectures for data processing. This system opens new challenges and allows exploring new reconstruction algorithm solutions, optimized on heterogeneous architecture and based on hardware-software co-design.

Eltec has proposed a camera and CT FPGA-based frame grabbers [11]; this device, called PCEY-0600 PC_EYE frame grabber, consists of multiple FPGAs and DRAM which is used for data buffering. These are interconnected on one single board.

Inspired by Xilinx and Intel-Altera ideas and challenges, we propose a Hardware/Software Co-Design architecture on SoC-FPGAs that merges different tasks inside one chip. We focus in this direction on improving the synchronization time between the different devices in CT and the timing to acquire and process the image data. Moreover, we avoid any external DRAM FIFO buffer with our pipeline architecture that directly transmits the filtered data through the gantry in the reconstruction system.

IV. ACQUISITION SYSTEM ARCHITECTURE

In the previous sections, we analysed the different acquisition systems and reconstruction architectures from the state-of-the-art. These have involved different medical and hardware requirements. In this section, we introduce our architecture based on the different tasks classified. Furthermore, we analyse their functionality and how they are distributed on hardware and software architectures. We have grouped all the tasks into three categories:

- *Safety controlling and synchronisation*
- *Data acquisition, buffering and re-transmission*
- *Image filtering and reconstruction*

The main *hardware components* described in Sec. II are interconnected and managed by slip-ring technology. Due the rotation of the gantry, we divide these components into two groups: *rotating* and *stationary* side modules. The former is fixed to the gantry and must rotate during the helical acquisition. The latter is fixed to the ground.

From hardware design point-of-view, this is an important classification, because the physical interconnection between the two groups can be only realised through *slip-ring technology* communication. This defines our communication bottleneck and also the design choices for the hardware requirements and protocols. Due to the presence of the magnetic field of the X-Ray and the rotation of the gantry, wireless data transmission is not possible in CTs. These technologies were investigated [4], but until now, they have not satisfied the safety and transmission performance requirements.

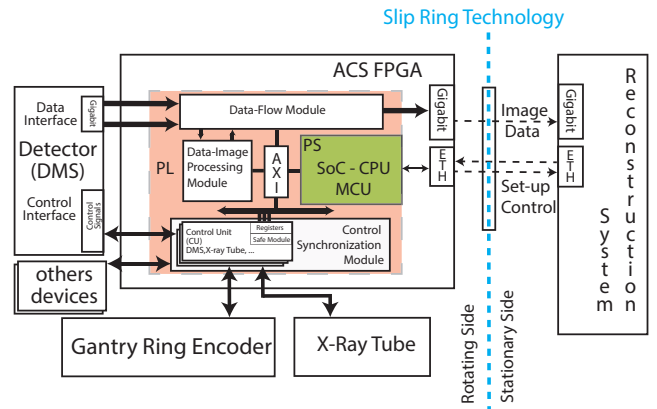


Fig. 2. CT-FPGA System Architecture

In our control and data acquisition system for CT, we use a SoC FPGA on the rotating side and another FPGA

on the Reconstruction System for collecting, processing and displaying data, as shown in Fig. 2. We focus on the Hardware/Software architecture of the *Acquisition-Control System FPGA (ACS)*, that is the core of our system. In our CT system, we have designed a multi-tier architecture with master/slave communication model between the different tiers, as shown in Fig. 3. In the multi-tier architecture, each (master) tier can only communicate with the tier below it, which is the slave in the protocol. The multi-tier architecture model is really important for characterising our design, because it allows the parametrization of the architecture for different slave devices, speed and custom data processing. It is scalable, and every module can be plugged inside and used as a new memory-mapped device. For managing the tasks above, we defined in our architecture three independent hardware/software modules that intercommunicate between each other; they are *Control Synchronisation Module*, *Data-Flow Module*, *Data-Image Processing Module*.

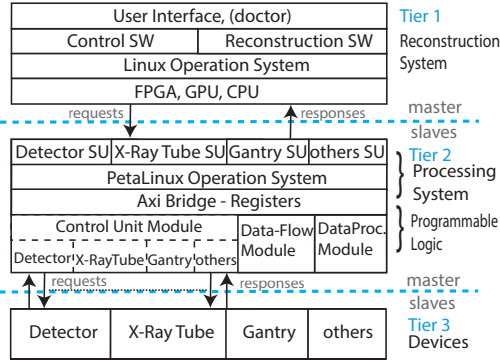


Fig. 3. Hardware/software multi-tier architecture

A. Control Synchronization Module

This module is responsible for handling the DMS, the X-ray tube and the gantry module. All these devices are normally designed with two independent control-interfaces: The former is the *software interface* that manages the device set-up. The latter is the *control interface* for real-time signals, and clock synchronisation. For each device and interface, we have implemented different units in the Programmable Logic (PL) of the ACS FPGA. These units are called *Software Units (SUs)* and the *Control Unit (CU)*; They are mapped to the Processing System (PS), via AXI-Lite registers, as a memory-mapped device, as shown in Fig. 2.

Through the SUs, the ACS FPGA is the “bridge” that manages user interface requests that are implemented on the stationary side system. The PS schedules the TCP/UDP requests and executes the related tasks on the PL. For the communication between the ACS FPGA and each device (e.g., DMS, X-ray tube), we have implemented custom protocols and transceivers inside the SU. Their job is to manage retransmission, error checking in hardware and software configuration of the PS. As mentioned above for the *control interface tasks* we design the CU that is independent of the SUs. As shown

in Fig. 2 and 3, there is one control module per device in the CU. This synchronises -directly in hardware- the different real-time signals for managing the X-ray Tube and the DMS acquisition with the gantry positioning and eventually real-time voltage errors and X-ray doses problems. In this way, the CU guarantees patient safety and avoids device damages.

B. Data-Flow Module

The Data-Flow Module manages the flow of the data from the DMS to the stationary side. It receives the data from the DMS via multi-channel optical connections. In the clinical CT during surgery, the data must arrive to the reconstruction system in the order of the milliseconds, with a speed of ~ 10 Gb/s. For this reason and unlike other architectures in literature, we can not store data in an external memory, rather we must use the DRAM available inside the FPGA for merging, buffering, checking the integrity and retransmitting the data to the reconstruction system. For the design of this module, the software is responsible for managing control flow setting (i.e., managing the number of protections, to acquire and handle eventual errors if during the acquisition there are too many corrupted data). If the errors are over the threshold, a hardware interrupt is issued and then handled by the software. Due to the required performance, we opted to use the PS as an asynchronous actor in the acquisition task, whereas, we manage and implement the protocol stack in the PL. The Data-Flow Module is designed as a flexible 5-stage pipeline architecture, controlled and configured by software on the PS:

- 1) *Transceiver stage*: it acquires the data from the physical optical channel, de-serialises and packet them;
- 2) *Data-link stage*: it implements the data-link layer protocol depending on the detector;
- 3) *Buffer stage*: it manages the collected data. We have one buffer for each physical channel associated;
- 4) *Merging stage*: it reads the data from the buffers and prepares them to be re-transmitted;
- 5) *Re-transmission*: it takes the ready data and transmits them to the reconstruction system;

For managing different configurations and behaviours of the pipe, we use a Control and a Status register mapped to the AXI-Lite registers. The software on the PS implements the different options for controlling the pipeline via registers.

C. Data-Image Processing Module

This module processes the streaming data from the buffering stage in the *Data-Flow Module*. In our architecture, the sampled data are stored, then processed on the PS side. It updates the data stored with the frequency of one projection (image) per second. In the meantime, the PS processes it, and updates the exposure acquisition time for improving the image quality of the successive projections. For storing the projection data the internal BRAM are used. This module is designed to be customised in future with different implementations and processing tasks, i.e. filtering, normalisation and reconstruction of the images. This module is connected to the *Merging stage* in the *Data-Flow Module* for transmitting the results to

the reconstruction system. The user can even decide to use the PS for transmitting the data via the TCP/UDP communication of the slip-ring technology.

V. IMPLEMENTATION AND EVALUATION

In the previous section, we described our architecture through the software and hardware tasks. For the evaluation, we explain how we implemented and integrated them inside a CT device, built in our laboratory. We integrated the architecture with the CT Detector CD300 64 ROW and the CT6000 X-ray tube from DUNLEE-Philips [8] with a gantry [6]. Based on this system, we will show the achieved performance in terms of area occupancy and data throughput of our ACS FPGA.

We have customized and implemented our architecture on "ZC706 Evaluation Board for the Zynq-7000 XC7Z045 SoC". For the implementation, we divided our system into hardware and software parts. The hardware part is described in SystemVerilog; it is scalable and parametrizable for using different internal clocks, buffer sizes and device modules with several detectors and X-Ray Tubes. We define in our architecture the parameters and the provided protocols from the selected devices. In the *Control Synchronization Module*, we used custom transceivers and different clocks based on the external clock system. In the *Data-Flow Module*, we selected the number of FIFOs and customized the *data-link-stage* based on the detector transmission protocol. We used a lookup table in the *data-link-stage* for parsing the packets in the protocol. The lookup table can be updated together with the protocol and the detector. In the implementation presented, we use the *Data-Image Processing Module* for collecting the images in the BRAM and for storing for the software processing on PS. In this first implementation, we are transmitting to the reconstruction system after the acquisition. We are implementing the communication between the ACS FPGA and the Reconstruction System, as explained in the architecture and shown in Fig. 2. The implementation of the reconstruction system is not presented in this work. In the software part, the software architecture is integrated with PetaLinux on "Zynq-7000 XC7Z045 SoC". The software architecture is entirely implemented in C, and it communicates with the hardware via AXI-Lite as *mmap* driver, and with the software user via Ethernet TCP/UDP socket. For the user, our ACS FPGA is an asynchronous system that responds for each request.

Resource	Utilization	Available	Utilization %
LUT	3305	218600	1.51
LUTRAM	133	70400	0.19
FF	4390	437200	1.00
BRAM	32	545	5.87
IO	28	362	7.73
GT	2	16	12.50
MMCM	1	8	12.50

Fig. 4. Resource utilization for Zynq-7000 XC7Z045 SoC

We synthesized and evaluated our system in terms of area and data transmission speed. The resource utilization of our architecture, reported in Fig. 4, requires less than 2% of the

available LUTs and less than 6% of the available BRAM. With the hardware/software co-design approach, we simplified the controlling mechanism and implemented it in software. In this way, the available area is used for the real-time signals, the critical safety tasks and the real-time data streaming. Furthermore, this area can be used for the enhancement of the Data-Image Processing Module. In terms of throughput, the transceiver is configured for 6.5 Gbit/sec, with different clock frequency up to 200 Mhz. This speed is required due to the amount of data that we receive in the system. The design is a pipelined-based architecture, it is flexible, and we can change the parameters. For this reason, we separate the streaming of data by the data processing and the software part for having support with higher speed transceiver and different detectors and X-ray tube in future.

VI. CONCLUSION

In this paper, we presented our hardware/software co-design architecture of control and data acquisition system for CT. We classified the tasks, and we partitioned them between hardware and software, based on the performance required and complexity of the component. We managed the controlling part on PS via software, and the data acquisition, synchronization of the critical real-time signals and data processing on the PL. In this way, we can manage the data within the time requested from the requirements of the CT, and we can use the available FPGA area for real-time image processing in the future. In fact, we can easily adapt the architecture to different CT devices and image processing application requests.

ACKNOWLEDGEMENT

The work of this paper is funded by the Federal Ministry of Education and Research within the project 'KIDs-CT' under grant number '13GW0229A'.

REFERENCES

- [1] Wesolowski, Jeffrey R., and Michael H. Lev. "CT: history, technology, and clinical aspects." Seminars in Ultrasound, CT and MRI. Vol. 26. No. 6. WB Saunders, 2005.
- [2] Daniele Passaretti, Jan Moritz Joseph and ThiloPionteck. Survey on FPGAs in Medical Radiology Applications: Challenges, Architectures and Programming Models, 2019 International Conference on Field-Programmable Technology (FPT). IEEE, 2019.
- [3] Faggioni L., Paolicchi F., and Neri E., Elementi di tomografia computerizzata. Vol.4. Springer Science & Business Media, 2011.
- [4] S. K. Yong and C. C. Chong, "An overview of multigigabit wireless through millimeterwave technology: Potentials and technical challenges," EURASIP J. Wireless Commun. Network., vol. 2007, no. 1, pp. 1–10, Jan.2007.
- [5] Xilinx, [https://www.xilinx.com/applications/medical/medical-imaging-ct-mri-pet.html#overview], [access:31/07/2019]
- [6] Schleifring, January 2018 "https://www.schleifring.de/wp-content/uploads/2018/01/CT-Applications_January18.pdf",
- [7] Altera Corporation, "Medical Imaging Implementation Using FPGAs" WP-MEDICAL-2.0, July 2010
- [8] Dunlee Philips, CD300 64 row detector, Datasheet detector
- [9] Parker, Michael. "Understanding peak floating-point performance claims." Technical White Paper WP-012220-1.0 (2014).
- [10] Giordano, R., S. Perrella, and D. Barbieri. "Radiation-Tolerant, High-speed Serial Link Design with SRAM-based FPGAs." arXiv preprint arXiv:1806.10677 (2018).
- [11] PCI Express Frame Grabber for Camera Link Cameras ELTEC systems, "https://www.eltec.de/pdf/datenblaetter/Datasheet_PC_EYE_CL.pdf" [access:20/01/2020]