

Research Article

Evaluation of the Reconfiguration of the Data Acquisition System for 3D USCT

Matthias Birk,¹ Clemens Hagner,¹ Matthias Balzer,¹ Nicole V. Ruiter,¹ Michael Hübner,² and Jürgen Becker²

¹*Institute for Data Processing and Electronics, Karlsruhe Institute of Technology, P.O. Box 3640, 76021 Karlsruhe, Germany*

²*Institute for Information Processing Technology, Karlsruhe Institute of Technology, P.O. Box 6980, 76049 Karlsruhe, Germany*

Correspondence should be addressed to Matthias Birk, matthias.birk@kit.edu

Received 27 August 2010; Accepted 10 February 2011

Academic Editor: Gilles Sassatelli

Copyright © 2011 Matthias Birk et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As today's standard screening methods often fail to diagnose breast cancer before metastases have developed, an earlier breast cancer diagnosis is still a major challenge. To improve this situation, we are currently developing a fully three-dimensional ultrasound computer tomography (3D USCT) system, promising high-quality volume images of the breast. For obtaining these images, a time-consuming reconstruction has to be performed. As this is currently done on a PC, parallel processing in reconfigurable hardware could accelerate both signal and image processing. In this work, we investigated the suitability of an existing data acquisition (DAQ) system for further computation tasks. The reconfiguration features of the embedded FPGAs have been exploited to enhance the systems functionality. We have adapted the DAQ system to allow for bidirectional communication and to provide an overall process control. Our results show that the studied system can be applied for data processing.

1. Introduction

Breast cancer is the most common type of cancer among women in Europe and North America. Unfortunately, in today's standard screening methods, breast cancer is often initially diagnosed after metastases have already developed [1]. The presence of metastases decreases the survival probability of the patient significantly. Thus, early breast cancer diagnosis is still a major challenge.

A more sensitive imaging method could allow for detection in an earlier state and thus enhance the survival probability. With this ultimate goal, we are researching and developing a three-dimensional ultrasound computer tomography (3D USCT) system for early breast cancer diagnosis [2]. This method promises reproducible volume images of the female breast fully in 3D.

Our initial measurements of clinical breast phantoms using the first 3D prototype showed promising results [3, 4] and led to a new and optimized aperture setup [5], currently built and shown in Figure 1. It is equipped with over 2000 ultrasound transducers. Further virtual positions of

the ultrasound transducers are created by rotational and translational movements of the complete sensor aperture.

In USCT, the interaction of unfocused ultrasonic waves with an imaged object is recorded from many different angles. During a measurement, the emitters sequentially send an ultrasonic wave front which interacts with the breast tissue and is recorded by the receivers as a pressure variation over time. These data sets, also called A-Scans, are sampled and stored for all possible emitter-receiver-combinations, resulting for our setup in over 3.5 million data sets and 20 GByte of raw data.

For acquisition of the A-Scan data during the measurement procedure, we use a massively parallel, FPGA-based data acquisition (DAQ) system. After completion, the recorded A-Scans are transferred to an attached computer workstation for the time-consuming image reconstruction. We exploit the pressure over time information in the A-Scans by a synthetic aperture focusing technique (SAFT) approach [6].

The necessary time for a volume reconstruction varies and strongly depends on both the desired image resolution



FIGURE 1: Image of the semiellipsoidal aperture of the new 3D USCT II. It is equipped with 628 ultrasound senders and 1413 receivers, grouped into 157 transducer array systems, mounted at the inner surface of the measurement basin.

and quality. However, our designated configuration for clinical application takes about 8 hours computed on an up to date PC. As we consider 5 minutes as an acceptable practical limit, the applied reconstruction algorithms need a significant acceleration of at least a factor 100 to be clinically relevant.

A promising approach to accelerate image reconstruction is parallel processing in reconfigurable hardware. In this work, we investigated the applicability of the DAQ system for further data processing tasks by a reconfiguration of the embedded FPGAs. As an exemplary processing sequence, we used the first step of the 3D USCT image reconstruction, which operates directly on the acquired A-Scans.

2. Related Work

The majority of processing algorithms which are applied in medical imaging methods feature an enormous compute and data intensity as well as a high degree of parallelism. Consequently, the application of parallel processors and further parallel accelerator architectures is investigated by many research groups. Special interest has recently been put on general purpose graphics processing units (GPUs), for example, [7–11], and the STI Cell Processor, for example, [11–13]. As our existing DAQ system is composed of a large number of FPGAs, which would be idle during the image reconstruction, the processing capabilities of such a system are here investigated.

Only a minority of medical imaging projects use FPGAs for computation. Most of them focus on well-established methods, like the X-ray-based Computed Tomography (CT) or Magnetic Resonance Imaging (MRI), which both differ

from the algorithms used for 3D USCT. In the following, a few examples are presented.

In [14], the most computational intensive part of the volume reconstruction in CT, that is, the filtering and back-projection step, is ported onto a reconfigurable platform. The authors chose an accelerator board with nine FPGAs as the target architecture and yield an acceleration factor of over 20, in comparison with the software-based approach. Although the reconfiguration capability of the FPGAs is mentioned in this work, it is not used within the application. Similar approaches targeting the backprojection algorithm in CT have been shown in [15–17], all of which producing a significant acceleration.

Furthermore, [18] describes a basic reconfigurable 16 channel front-end for magnetic resonance imaging (MRI) on one FPGA. By means of the created partial dynamic reconfiguration framework, the authors replace the data acquisition module after the actual measurement took place by various processing units and therefore the chip resources are reused for image reconstruction. As this framework is created as a proof-of-concept, an acceleration factor is not given. The presented reconfiguration framework is very interesting as it follows the same basic principle as our work; however, the performed partial reconfiguration is not possible with the Altera FPGAs embedded in our DAQ system.

Closest to our application is a novel ultrasound based method presented in [19]. Jensen et al. show an FPGA-based data acquisition and processing system for ultrasound synthetic aperture imaging. The overall system is composed of 320 FPGAs, distributed over 64 identical boards, and is able to process 1024 ultrasound signals in parallel. However, the embedded FPGAs are statically configured and a dynamic reconfiguration of the FPGAs is not considered.

In summary, to the best of our knowledge, no reconfigurable computing system based on the system-wide FPGA reconfiguration for SAFT-based medical image reconstruction has been introduced.

3. Data Acquisition System

The data acquisition (DAQ) system has been developed as a common platform for multiproject usage, for example, in the Pierre Auger Observatory [20], the Karlsruhe Tritium Neutrino Project [21] and has also been adapted to the needs of 3D USCT. The DAQ system is described in detail in the following subsections.

3.1. Setup and Functionality. In the USCT configuration, the DAQ system consists of 21 expansion boards: one second level card (SLC) and 20 identical first level cards (FLC). The complete system fits into one 19" crate, which is depicted in Figure 2. The SLC is positioned in the middle between 10 FLCs to the right and left.

The DAQ system holds 81 Altera Cyclone II FPGAs. Table 1 gives an overview of the FPGAs' device features. In total, up to 480 receiver signals can be acquired concurrently by assigning 24 channels to each FLC. This results in a receiver multiplex-factor of three for the acquisition of all



FIGURE 2: Image of the DAQ system in the USCT configuration. It is composed of one Second-Level Card (SLC) for measurement control and communication management (middle slot) and 20 First-Level Cards (FLC) for parallel sensor signal acquisition and data storage.

TABLE 1: FPGA device elements [22].

Element	Number per FPGA
Logic elements	33 216
Embedded multipliers (9 bit)	70
Total memory bits	483 840

possible emitter-receiver combinations in the new 3D USCT prototype.

The SLC controls the overall measurement procedure. It triggers the emission of ultrasound pulses and handles data transfers to the attached reconstruction PC. It is equipped with one Cyclone II FPGA and a processor module (Intel CPU, 1 GHz, 256 MB RAM) running a Linux operating system. Communication with the attached PC is either possible via Fast Ethernet or an USB interface. For communication between SLC and the FLCs within the DAQ system, a custom backplane bus is used.

3.2. First Level Card. An FLC consists of an analogue and a digital part. However, only the digital part will be considered throughout this paper. A block diagram of this part is given in Figure 3. In addition to three 8-fold ADCs for digitization of the 24-assigned receiver channels, an FLC is equipped with four Cyclone II FPGAs: we use one FPGA as a local control instance (Control FPGA, Cntrl FPGA). It handles communication and data transfer to the other FPGAs on board and to the SLC via backplane bus. We employ the other three FPGAs for actual signal acquisition. Each of these is fed by one ADC and thus processes eight receiver channels in parallel.

There are two different types of memory modules as intermediate storage for the acquired A-Scans on-board available: each Comp FPGA is connected to a distinct static RAM module (QDR II, 2 MB each) and the control FPGA

is attached to a dynamic RAM module (DDR II, 2 GB), summing up to a system capacity of over 40 GB.

There are two separate means of communication between the control FPGA and the computing FPGAs (see Figure 3): a slow local bus with a width of 32 bit (Local Bus, 80 MB/s) and 8-bit wide direct data links (Fast Links, 240 MB/s per computing FPGA). Additionally, there are several connections for synchronization between the control FPGA and the computing FPGAs on each board.

4. Methodology

As outlined in Section 1, 3D USCT promises high-quality volumetric images of the female breast and therefore has a high potential in cancer diagnosis. However, it requires time-consuming image reconstruction steps, limiting the method's general applicability.

To achieve a clinical relevance of 3D USCT, that is, application in clinical routine, image reconstruction has to be accelerated by at least a factor of 100. A promising approach to reduce overall computation time is parallel processing of reconstruction algorithms in reconfigurable hardware.

In the current design, we use the DAQ system only to control the measurement procedure and to acquire the ultrasound receiver signals. In this evaluation, we investigated the utilization of the FPGAs in the DAQ system for further processing tasks. Due to limited FPGA resources, the full set of the necessary processing algorithms, that is, for data acquisition, signal processing, and image reconstruction, cannot be configured statically onto the FPGAs at the same time.

Therefore, a reconfiguration of the FPGAs is necessary to switch between different configurations, enabling signal acquisition and further processing on the same hardware system. As the DAQ system has not been designed for further processing purposes, the scope of this work was to identify its capabilities as well as architectural limitations in this regard.

Only a reconfiguration of the FPGAs on the FLCs has been investigated since these hold the vast majority of FPGAs within the complete system. Therefore, only these cards and their data-flow are considered in the following sections. Furthermore, an interaction or data exchange between different FLCs has not been considered in this study.

The hardware setup of an FLC is given in Section 2 and shown in Figure 3. The detailed data-flow of an FLC in conventional operation mode is shown in Figure 4. As 24-receiver channels are processed per FLC, the signals are split into groups of eight. Every group is digitized in one ADC and fed into one computing FPGA. Within an FPGA, the signals are digitally filtered and averaged with previous recordings of the same emitter-receiver combination by means of the attached QDR memory. Note that during DAQ, the same emitter-receiver combination is sampled multiple times in order to improve the signal-to-noise ratio. Finally, the measurement data is transmitted via fast data links to the control FPGA, where it is stored in DDR II memory. After the complete measurement is finished, the resulting data is

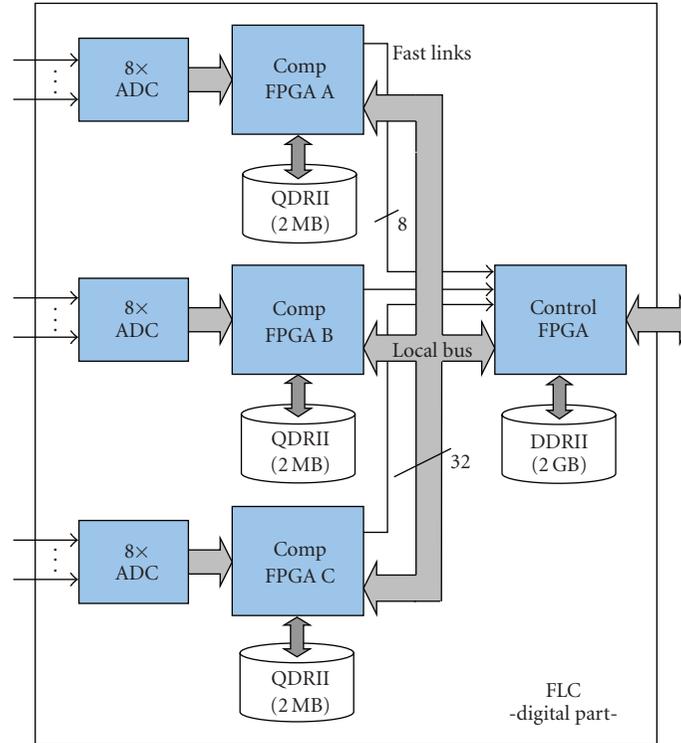


FIGURE 3: Block diagram of the digital part of an FLC in the 3D USCT DAQ system. It is equipped with four Altera Cyclone II FPGAs, one for local control (control FPGA, Cntr FPGA) and three for signal acquisition (computing FPGAs, Comp FPGA). Each Comp FPGA is fed by an 8-fold ADC and is attached to a 2 MB QDR static RAM module. The Cntrl FPGA is attached to a 2 GB DDRII dynamic RAM. Communication on each board is either possible by the slower local bus (Local Bus, 80 MB/s) or by fast data links (Fast Link, 240 MB/s).

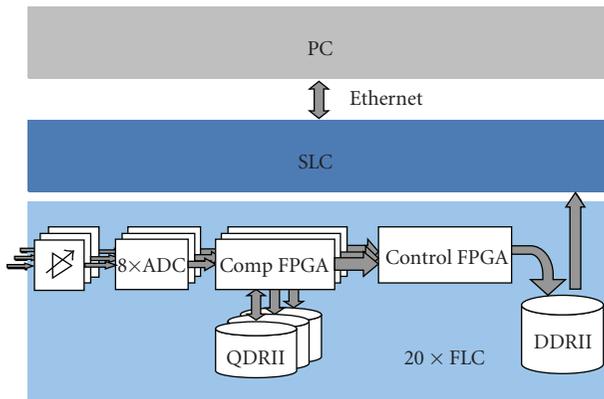


FIGURE 4: Detailed data-flow of one FLC during the conventional acquisition mode: Every FLC processes 24 receiver channels in parallel, whereas a group of eight signals is digitized in a single ADC. The digital signals are filtered and averaged in the computing FPGAs. Finally, the signals are transmitted to the Control FPGA and stored in DDRII memory.

transferred via backplane bus to the SLC and further to the attached PC for signal processing and image reconstruction.

In this work, we reconfigured the FPGAs after completion of a measurement cycle, that is, when the data is stored in DDRII memory, and thus switched from conventional

acquisition to data processing mode. As depicted in Figure 5, instead of transmitting the data sets via SLC to the attached PC, we loaded them back to QDR II and subsequently processed them in the computing FPGAs. After completion, we transmitted the resulting data back to the control FPGA and again stored it in DDRII memory. For providing this reconfiguration methodology, we had to perform the following tasks:

- (i) preventing data loss during reconfiguration,
- (ii) establishing communication and synchronization,
- (iii) implementing bidirectional communication interfaces.

4.1. Preventing Data Loss during Reconfiguration. Our DAQ system is built up of Altera Cyclone II FPGAs, which do not allow a partial reconfiguration [22]. Therefore, we had to reconfigure the complete FPGA chip. To prevent a loss of measurement data during the reconfiguration cycle, all data has to be stored outside the FPGAs in on-board memory, that is, QDRII or DDRII memory.

The QDRII is static memory so that stored data is not corrupted during reconfiguration of the FPGAs. However, only the larger memory (DDRII) is capable of holding all the data sets recorded per FLC. This dynamic memory module needs a periodic refresh cycle to hold stored data. On the FLC, the control FPGA is responsible for triggering

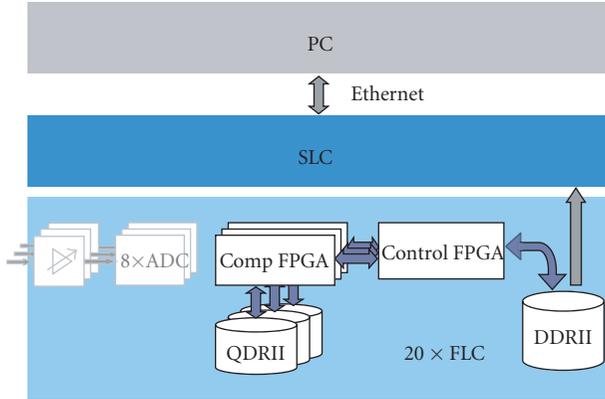


FIGURE 5: Detailed data-flow of one FLC during the newly created processing mode: as the data sets were previously stored in DDRII memory, we transferred them back to QDRII memory and processed them in the computing FPGAs. Finally, we stored the resulting data again in DDRII memory.

these refresh cycles. During a reconfiguration, this FPGA would not be able to perform this task. Since a refresh interval of the dynamic memory module is in the order of a few microseconds and a reconfiguration of the control FPGA takes even in the fastest mode about 100 ms [22], we had to ensure that it is not reconfigured; otherwise, data in the DDRII memory would be lost. Due to this requirement, we were only able to reconfigure the three computing FPGAs during operation.

At a normal startup of the DAQ system, all FPGAs on an FLC are configured via passive serial mode [22] with configuration data provided by an embedded ROM. As depicted in Figure 6, first the control FPGA and then all three computing FPGAs are configured in parallel in this mode. Constraint by the FLC printed circuit board, in current hardware setup we were not able to exclude the control FPGA from a configuration in passive serial mode. Therefore, in order to address and reconfigure each FPGA on the FLC separately, we had to use the JTAG configuration mode [22]. The JTAG chain through all four FPGAs is shown in Figure 7: each FPGA within the chain is configured sequentially with configuration data from an external programmer (PC).

4.2. Communication and Synchronization. Another important task in establishing the described reconfiguration methodology was to organize both communication and control on the FLC and furthermore, the synchronization of the parallel processing on the computing FPGAs.

As described in Section 2, there are two means of communication between the computing FPGAs and the control FPGA (see Figure 3): the slow local bus (Local Bus) and fast direct data links (Fast Links). In conventional DAQ operation mode, measurement data is transmitted only in the direction from the computing FPGAs to the control FPGA. Due to operational constraints in the FPGA pins, which are used for the fast links, this connection can only be operated in the above mentioned sense, that is, unidirectional. Thus, in

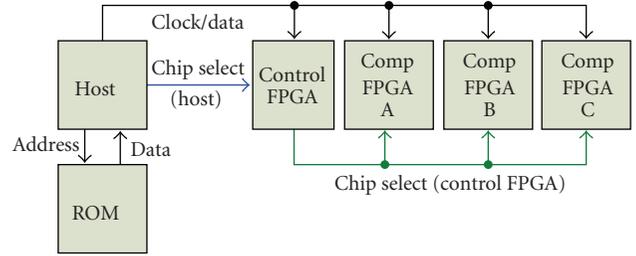


FIGURE 6: Passive serial configuration of the FPGAs on an FLC at system startup time: firstly, the control FPGA and after that the computing FPGAs are configured in parallel with configuration data from an embedded ROM.

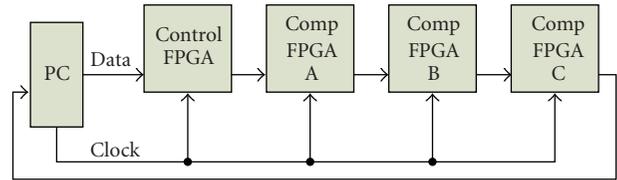


FIGURE 7: JTAG chain for reconfiguration of the FPGAs on an FLC. In JTAG reconfiguration mode, each FPGA can be selected separately for a reconfiguration, whereas deselected FPGAs remain in normal operation mode. Reconfiguration of different FPGAs within the chain is sequential.

the created processing mode, we had to rely on the slower local bus for data transfer since only this connection allows a bidirectional communication. The complete communication infrastructure is shown in Figure 8.

As the control FPGA is not reconfigured during operation, we had to statically configure it to handle data transfers appropriately in each system state, that is, DAQ and processing mode. Therefore, it must be able to determine the current state. As also depicted in Figure 8, we used a single on-board spare connection (*conf_state*) for that purpose, and therefore connected *conf_state* to all four FPGAs. In addition, we established process control and synchronization by further point-to-point links. Thus, each computing FPGA can be addressed and selected directly by the control FPGA: the respective *chip_select* signal triggers processing in a computing FPGA and completion of processing is indicated to the control FPGA by the *busy* signal.

4.3. Communication Interfaces. A further task was structuring communication and memory interfaces in the computing FPGAs. As a result, we created modular interfaces for transmitting data over the Local Bus (communication I/F) and storing data in QDRII memory (memory I/F). Figure 9 shows a block diagram of these modules on the computing FPGAs. This modular design allows a simple exchange of algorithmic modules without the need to change further elements.

The communication interface is managed by the control FPGA. It performs data transfers via Local Bus during processing or via fast data links during DAQ mode. The

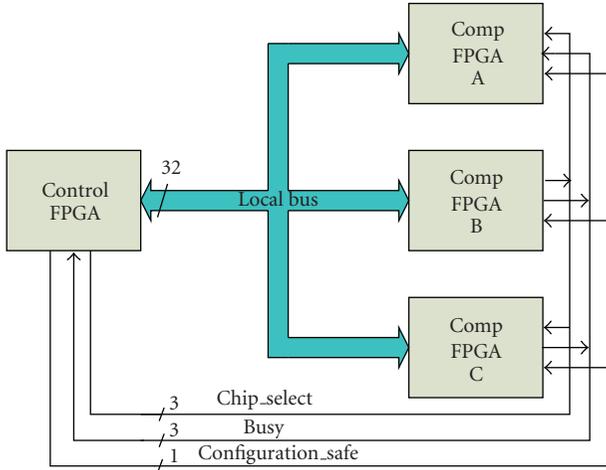


FIGURE 8: Communication structure during processing mode on an FLC: bidirectional data transfer is only possible via the slower Local Bus (80 MB/s). Separate point-to-point links (*chip_select* & *busy*) are used for control and synchronization of parallel processes. A further single point-to-point link is connected to all four FPGAs, indicating the current system state, that is, DAQ or processing mode.

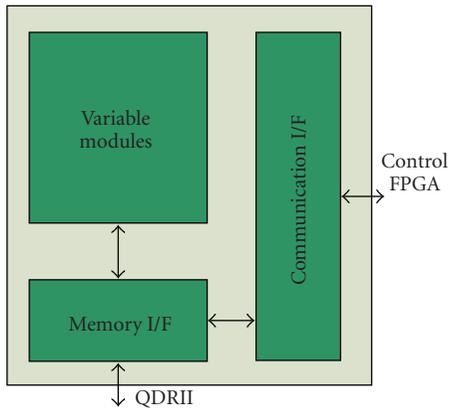


FIGURE 9: Block diagram of a computing FPGA: generic interfaces for communication with the control FPGA (communication I/F) via Local Bus or Fast Links and access to QDRII memory (memory I/F). The variable algorithmic part is also indicated.

memory interface handles accesses to the QDRII memory. We can either access it by the control FPGA via Local Bus or from the algorithmic modules. In the current configuration, an algorithmic module only interacts with the memory interface and thus, only processes data which has already been stored in QDRII memory.

In order to ensure a seamless data transfer over the Local Bus, we supplemented the respective memory interface in the control FPGA by a buffered access mode to the DDRII memory. When we initialize a data transfer, enough data words are preloaded into a buffer so that the transmission is not interrupted during a refresh cycle.

5. Experimental

We tested the reconfigurable computing system by acquisition of test pulses, followed by the reconfiguration of the computing FPGAs and an exemplary data processing. The used pulse was in the same frequency range as regular measurement data and was handled like a normal data set (A-Scan). Our main goals of this test were to determine the required transfer time per data set over the Local Bus and the reconfiguration times per FPGA and per FLC.

5.1. Test Setup. For functional validation and performance measurements, we used a reduced setup of our DAQ system, which contains the SLC and only one FLC. However, as our exemplary processing sequence operates only locally on the FLC and does not include communication or interactions with other FLCs, this setup allows us an extrapolation of both reconfiguration and processing times for a fully equipped DAQ system.

5.2. Detailed Test Procedure. We tested the system as follows: at system startup, we loaded the initial DAQ configuration into the FPGAs as outlined in Section 4.1. Afterwards, we applied the test pulse at the inputs of the ADCs on the FLC. In DAQ mode, the digitized pulse is finally stored in DDRII memory on the FLC.

The further detailed procedure is indicated in Figure 10. After our manual reconfiguration of the computing FPGAs via JTAG, we transferred the first set of A-Scans to a computing FPGA (FPGA A) via Local Bus, stored them in its attached QDRII memory and subsequently processed them. While data in this FPGA is being processed, we supplied the other two computing FPGAs (FPGA B and FPGA C) with their first sets of A-Scans and started processing on these FPGAs. After processing is completed in FPGA A, we transmitted the resulting A-Scan data back to DDRII memory and sent further unprocessed A-Scans to this FPGA. We repeatedly applied this scheme until all A-Scans had been processed.

5.3. Exemplary Processing Sequence. We used for this test a basic version of the so-called adapted matched filtering [23]. This processing sequence is applied as the first step of the 3D USCT image reconstruction and operates directly on the raw A-Scans in order to improve the signal-to-noise ratio, resulting in an enhanced image contrast. All processing steps are performed separately and independently from each other on all acquired A-Scans.

Initially, each A-Scan consists of 3000 time discrete samples with a width of 16 bit. Due to resource limitations, we had to retain this signal width throughout the processing and thus had to reduce the bit width after each computational step. The complete processing chain as implemented on the computing FPGAs and executed in this test is depicted in Figure 11.

Firstly, we read the raw A-Scan from QDRII memory and wrote it into embedded memory blocks within the FPGA. Then, we correlated this A-Scan with the matched filter [24]

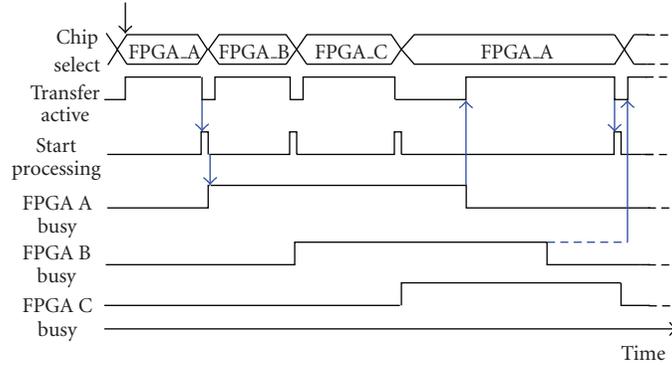


FIGURE 10: Test procedure after DAQ and manual reconfiguration of the Comp FPGAs via JTAG. Firstly, we supplied computing FPGA A with a set of A-Scans and started processing on this FPGA. While processing is underway, we initiated data transfer and processing on the Comp FPGAs B and C. After completion of processing on FPGA A, we transferred the resulting data back to DDRII memory and loaded further unprocessed A-Scans. This scheme is repeatedly applied.

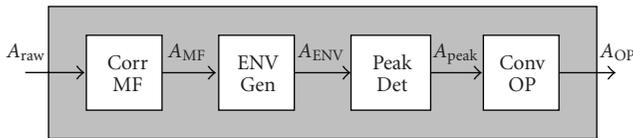


FIGURE 11: Complete processing chain of the adapted matched filtering [19] as implemented on the computing FPGAs. Firstly, we correlated the raw A-Scan with the matched filter signal (CorrMF). Then, we generated the absolute signal envelope (EnvGen) and in the following reduced it to its local maxima (PeakDet). Finally, the intermediate signal is convoluted with an optimal pulse for reflectivity imaging (ConvOP).

(CorrMF). This is the expected wave form at the receivers and was estimated by previous empty measurements. The correlation kernel consists of 64 samples of 12-bit integers. We performed this calculation by means of embedded multipliers [22].

In the next step, we generated the envelope of the resulting signal (EnvGen) by firstly creating the absolute value signal and then applying an adapted cascaded integrated comb (CIC) filter [25]. Subsequently, we reduced this intermediate signal to its local maxima (PeakDet), that is, we retained the signal values at local peaks, whereas all other samples within the A-Scan are set to zero. We did this in a streamed manner by a direct comparison of every sample with both its immediate predecessor and successor.

Finally, we convoluted this signal with an optimal pulse for reflectivity imaging as illustrated in [26]. The convolution kernel consists of 17 samples of 12-bit integers. Again, we performed the calculation by means of the embedded multipliers. After completion of the described processing steps, we wrote the resulting processed A-Scans back to the attached QDRII memory module.

5.4. Results. A JTAG configuration of a single computing FPGA requires 1.8 s, resulting in a reconfiguration time of 5.4 s for one FLC, when only the three computing FPGAs are configured in the JTAG chain. A reconfiguration of all 60

TABLE 2: JTAG reconfiguration times.

Procedure	Required time
Reconf. of one computing FPGA	1.8 s
Reconf. of one FLC	5.4 s
Extrapolated reconf. of the DAQ system	~2 min

TABLE 3: Computing FPGA occupation.

Components	Configuration		
	Data acquisition	Data processing	Comm & mem I/F only
Logic elements	17%	19%	4%
Embedded multipliers	97%	91%	0%
Memory bits	<1%	68%	0%

computing FPGAs, distributed over the 20 FLCs in the complete DAQ system, would take up to 2 minutes by building up a JTAG chain through all FPGAs. The determined JTAG reconfiguration times are illustrated in Table 2.

The transfer of one data set via Local Bus in either direction, that is, from control FPGA to computing FPGA or vice versa, takes $75 \mu\text{s}$. Usage of the Local Bus is limited to one computing FPGA at a time and the same bus is used for data transfer to and from all three computing FPGAs.

Table 3 outlines the occupation of the computing FPGA during the test procedure. The extensive use of embedded multipliers in DAQ mode, which are required due to the hard real-time constraints, states a clear demand for the established reconfiguration methodology. Furthermore, the implemented communication and memory interfaces occupy only 4% of the device's logic elements.

The implemented adapted matched filtering takes $450 \mu\text{s}$ per A-Scan on a single computing FPGA. By employing the entangled processing method in Figure 10, the processing time of a set of three A-Scans equals $750 \mu\text{s}$ on one FLC. As this processing sequence does not include any interactions with other FLCs, the extrapolated processing time per A-Scan

on the complete DAQ system (20 FLCs) results in to $12.5 \mu\text{s}$. The current computation in Software on an Intel Core i7-920 at 2.67 GHz takes in Matlab about $840 \mu\text{s}$ and for a multi-threaded C implementation using 8 threads (hyperthreading), compiled with gcc 4.4 with $-O3$ flag, about $40 \mu\text{s}$ per A-Scan.

6. Discussion and Conclusions

In this paper, we presented a feasible concept of a reconfigurable computing system based on an existing DAQ system for 3D USCT. As the main result, we showed the possibility of reusing this system for data processing.

The main goals were to analyze the DAQ system's characteristics, to determine limitations and derive implementations strategies. Thus, the performance comparison of our exemplary processing sequence may be misleading. The algorithmic implementation could be further optimized in both computational strategy and quality of results. Nevertheless, already in this basic version, we obtained an acceleration of about factor three if we compare the pure computational time with a multithreaded C implementation on an up to date CPU. However, when comparing the total processing time for a full set of A-Scans (~ 3.5 million) and taking the necessary reconfiguration of the complete DAQ system into account, the achieved speed-up almost balances out. On the other hand, File I/O from and to hard disk has not been considered in the software-based processing time, which also contributes substantially and degrades the overall performance by a factor of two.

The main drawback of the current system is the long reconfiguration time. Thus, the reconfiguration cycles impact the total processing time significantly. To which extent this constraint will restrict the applicability of the presented method needs further investigation. However, the reconfiguration time can be reduced by a factor of 20 by separate JTAG chains for each FLC and a concurrent reconfiguration.

Likewise, due to the slow bidirectional data transfer over the Local Bus, the achievable performance during the processing phase is also limited. This issue could be improved by a modified communication scheme where data from the computing FPGAs to the control FPGA is transferred via Fast Links and the Local Bus is only used for the opposite direction from the control FPGA to the computing FPGAs. As the Fast Links have a much larger data rate, this would accelerate the overall data transfer by a factor two with the Local Bus being the limiting factor.

Assuming the applied data parallel processing strategy, that is, each computing FPGA performs the same computation on a different data set, a high efficiency can be reached if the following condition holds: the parallelized processing time per A-Scan on a computing FPGA has to be longer than $450 \mu\text{s}$, which is six times the transfer time of a single data set. In this context, parallelized time is the processing time per A-Scan on one computing FPGA divided by the number of concurrently processed A-Scans on this FPGA. Then, the transfer times to and from all three FPGAs could be hidden.

If this processing strategy is not feasible for a given algorithm or it requires communication between different FLCs other effects come into play. These may limit scalability and need further consideration.

7. Outlook

For future work, two obvious aspects have been derived in the last section. Namely, reducing data transfer time by a modified communication scheme in the processing phase and reducing reconfiguration time by parallel JTAG chains.

A further task will be porting more processing algorithms to the DAQ system in order to evaluate the established reconfiguration ability in the real application. This also includes the operation of the full DAQ system with all 20 FLCs. So far, we did not consider a direct communication between the computing FPGAs on an FLC as well as an interaction of different FLCs in general. This will open up manifold implementation strategies for algorithmic modules besides the applied data parallel processing scheme, but also increases the implementation effort.

In the long term, a redesign of the DAQ system will focus on enhanced processing capabilities. This includes high-speed data transfer as well as research in heterogeneous computing concepts by combining FPGAs with further processing elements like GPUs or multicore CPUs.

Acknowledgments

The authors acknowledge support by Deutsche Forschungsgemeinschaft and Open Access Publishing Fund of Karlsruhe Institute of Technology.

References

- [1] D. van Fournier, H. J. H.-W. Anton, and G. Bastert, "Breast cancer screening," in *Cancer Diagnosis: Early Detection*, P. Bannasch, Ed., pp. 78–87, Springer, Berlin, Germany, 1992.
- [2] H. Gemmeke and N. V. Ruiter, "3D ultrasound computer tomography for medical imaging," *Nuclear Instruments and Methods in Physics Research A*, vol. 580, no. 2, pp. 1057–1065, 2007.
- [3] N. V. Ruiter, G. F. Schwarzenberg, M. Zapf, and H. Gemmeke, "Conclusions from an experimental 3D ultrasound computer tomograph," in *Proceedings of IEEE Nuclear Science Symposium Conference Record (NSS/MIC '08)*, pp. 4502–4509, October 2008.
- [4] N. V. Ruiter, G. F. Schwarzenberg, M. Zapf, A. Menshikov, and H. Gemmeke, "Results of an experimental study for 3D ultrasound CT," in *Proceedings of NAG-DAGA International Conference on Acoustics*, 2009.
- [5] G. F. Schwarzenberg, M. Zapf, and N. V. Ruiter, "Aperture optimization for 3D ultrasound computer tomography," in *Proceedings of IEEE Ultrasonics Symposium (IUS '07)*, pp. 1820–1823, October 2007.
- [6] S. R. Doctor, T. E. Hall, and L. D. Reid, "SAFT—the evolution of a signal processing technology for ultrasonic testing," *NDT International*, vol. 19, no. 3, pp. 163–167, 1986.
- [7] H. Scherl, B. Keck, M. Kowarschik, and J. Hornegger, "Fast GPU-based CT reconstruction using the Common Unified

- Device Architecture (CUDA),” in *Proceedings of IEEE Nuclear Science Symposium Conference*, vol. 6, pp. 4464–4466, 2007.
- [8] K. Mueller, F. Xu, and N. Neophytou, “Why do commodity graphics hardware boards (GPUs) work so well for acceleration of computed tomography?” in *Computational Imaging V*, vol. 6498 of *Proceedings of the SPIE*, San Jose, Calif, USA, January 2007.
- [9] S. S. Stone, J. P. Haldar, S. C. Tsao, W. M. W. Hwu, B. P. Sutton, and Z. P. Liang, “Accelerating advanced MRI reconstructions on GPUs,” *Journal of Parallel and Distributed Computing*, vol. 68, no. 10, pp. 1307–1318, 2008.
- [10] T. Schiwietz, T.-C. Chang, P. Speier, and R. Westermann, “MR image reconstruction using the GPU,” in *Medical Imaging 2006: Physics of Medical Imaging*, vol. 6142 of *Proceedings of SPIE*, February 2006.
- [11] S. Van Der Maar, K. J. Batenburg, and J. Sijbers, “Experiences with cell-BE and GPU for tomography,” in *Proceedings of the 9th International Workshop on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS '09)*, vol. 5657 of *Lecture Notes in Computer Science*, pp. 298–307, July 2009.
- [12] M. Knaup, S. Steckmann, O. Bockenbach, and M. Kachelrieß, “Tomographic image reconstruction using the Cell Broadband Engine (CBE) general purpose hardware,” in *Computational Imaging V*, vol. 6498 of *Proceedings of SPIE*, January 2007.
- [13] H. Scherl, M. Koerner, H. Hofmann, W. Eckert, M. Kowarschik, and J. Hornegger, “Implementation of the FDK algorithm for cone-beam CT on the cell broadband engine architecture,” in *Medical Imaging 2007: Physics of Medical Imaging*, vol. 6510 of *Proceedings of SPIE*, February 2007.
- [14] B. Heigl and M. Kowarschik, “High-speed reconstruction for C-arm computed tomography,” in *Proceedings of the 9th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, 2007.
- [15] N. Sorokin, “Parallel backprojector for cone-beam Computer Tomography,” in *Proceedings of the International Conference on Reconfigurable Computing and FPGAs (ReConFig '08)*, pp. 175–180, December 2008.
- [16] S. Coric, M. Leaser, E. Miller, and M. Trepanier, “Parallel-beam backprojection: an FPGA implementation optimized for medical imaging,” in *Proceedings of the 10th ACM International Symposium on Field-Programmable Gate Arrays (FPGA '02)*, pp. 217–226, February 2002.
- [17] N. Gac, S. Mancini, and M. Desvignes, “Hardware/Software 2D-3D backprojection on a SoPC Platform,” in *Proceedings of the ACM Symposium on Applied Computing*, pp. 222–228, April 2006.
- [18] I. L. Dalal and F. L. Fontaine, “A reconfigurable FPGA-based 16-channel front-end for MRI,” in *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, pp. 1860–1864, 2006.
- [19] J. A. Jensen, M. Hansen, B. G. Tomov, S. I. Nikolov, and H. Holten-Lund, “System architecture of an experimental synthetic aperture real-time ultrasound system,” in *Proceedings of IEEE Ultrasonics Symposium (IUS '07)*, pp. 636–640, October 2007.
- [20] H. Gemmeke et al., “First measurements with the auger fluorescence detector data acquisition system,” in *Proceedings of the 27th International Cosmic Ray Conference*, 2001.
- [21] A. Kopmann, T. Bergmann, H. Gemmeke et al., “FPGA-based DAQ system for multi-channel detectors,” in *Proceedings of IEEE Nuclear Science Symposium (NSS/MIC '08)*, pp. 3186–3190, October 2008.
- [22] Altera Corporation, *Cyclone II Device Handbook*, Altera, 2007.
- [23] N. V. Ruiter, G. F. Schwarzenberg, M. Zapf, and H. Gemmeke, “Improvement of 3D ultrasound computer tomography images by signal pre-processing,” in *Proceedings of IEEE International Ultrasonics Symposium (IUS '08)*, pp. 852–855, November 2008.
- [24] D. O. North, “An Analysis of the factors which determine signal/noise discrimination in pulsed-carrier systems,” *Proceedings of the IEEE*, vol. 51, no. 7, pp. 1016–1027, 1963.
- [25] E. B. Hogenauer, “An economical class of digital filters for decimation and interpolation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 155–162, 1981.
- [26] S. J. Norton and M. Linzer, “Ultrasonic reflectivity tomography: reconstruction with circular transducer arrays,” *Ultrasonic Imaging*, vol. 1, no. 2, pp. 154–184, 1979.