

# Real Time Floating Point SAR Focusing on FPGA

Srikanth Mandapati<sup>a</sup>, Ulrich Balss<sup>a</sup>, and Helko Breit<sup>a</sup>

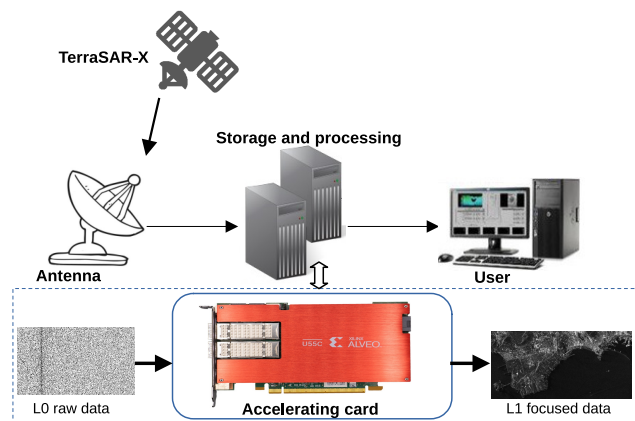
<sup>a</sup>German Aerospace Center (DLR), Remote Sensing Technology Institute, SAR Signal Processing, 82234 Weßling, Germany (e-mail: srikantha.mandapati@dlr.de; ulrich.balss@dlr.de; helko.breit@dlr.de)

## Abstract

Due to the demand for faster processing times, less required space, and efficient power consumption in the era of high-performance computing (HPC) as well as onboard processing, the use of Graphics Processing Units (GPUs) and in particular Field-Programmable Gate Arrays (FPGAs) have emerged as powerful and versatile solutions for accelerating data workloads. Offloading compute-intensive workloads from the Central Processing Unit (CPU) to these accelerating cards enables faster processing, analysis of large data, and resulting in improved time-to-insight and decision-making capabilities. Specifically, FPGAs can deliver exceptional performance, energy efficiency and flexibility in contrast to GPUs. By taking these advantages of the FPGAs on the data accelerator cards, the Synthetic Aperture Radar (SAR) image focusing algorithm is implemented on an AMD Xilinx Alveo U55C card. The processor is developed in floating-point on the FPGA to achieve higher accuracy and can focus an area covering 375 km<sup>2</sup> in approximately 1.5 s.

## 1 Introduction

SAR processors have to focus a large amount of raw data within a reasonable timespan. Thus, the adequate choice of hardware to be used plays an important role in system design. In the recent years, several computing platforms, including CPUs [1], Digital Signal Processors (DSPs) [2], GPUs [3], and FPGAs or a combination of FPGA with ASIC/CPU [4, 5, 6] have been explored and utilized extensively for SAR focusing either on-board or ground. Each of these platforms have their own strengths, consideration and limitations. CPUs are general-purpose processors that excel at handling and executing of sequential processing tasks. However, SAR focusing involves fast Fourier transforms (FFTs) and matrix multiplications, which can be time consuming on CPUs due to their limited parallel processing capabilities. GPUs are highly parallel processors designed specifically for rendering graphics and performing massive parallel computations by executing multiple threads simultaneously on thousands of cores. But GPUs consume more power than CPUs or FPGAs. Though, DSPs are specifically designed and optimized to handle real-time signal processing tasks with lower power consumption, they have limited I/O options and less flexibility compared to FPGAs. On the other side, FPGAs are reconfigurable hardware devices that can be reprogrammed to implement specific tasks efficiently. They offer high flexibility and can be tailored to optimize SAR focusing algorithms, resulting in improved performance and energy efficiency compared to CPUs and GPUs. FPGAs are particularly advantageous with regard to their low power consumption when low-latency and real-time processing are crucial requirements. They can achieve high throughput and deterministic performance. Despite of having many advantages, FPGAs are expensive and require specialized knowledge and expertise to program effectively, as they use Hardware

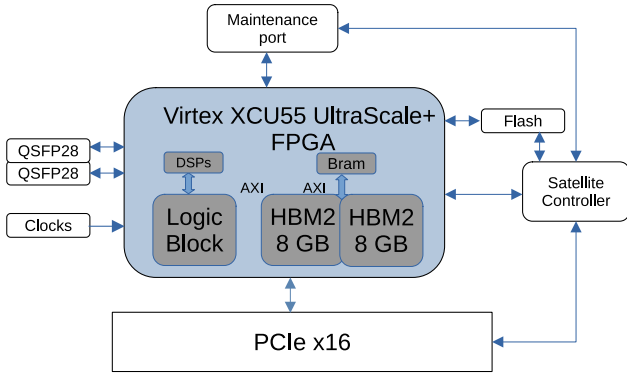


**Figure 1** The basic overview of SAR focusing flow at ground station.

Description Languages (HDL) such as Verilog or VHDL. This makes programming FPGAs more complex and time-consuming compared to programming GPUs or CPUs using high-level languages like C++ or Python. High Level Synthesis (HLS) is not utilized because, at the time being, it does not satisfyingly cope with the demands of an efficient hardware implementation of such a complex algorithm like SAR focusing.

## 2 Background and motivation

In the traditional SAR processing chain, the radar captures the scene of interest and then raw data is downlinked via high rate communication links to the ground station where the data is processed. The processing on CPUs consumes considerable amount of time and energy. The utilization of FPGAs is an efficient alternative and overcomes these drawbacks. In the EO-ALERT project [4], we already in-



**Figure 2** Block diagram of Alveo U55C accelerator card

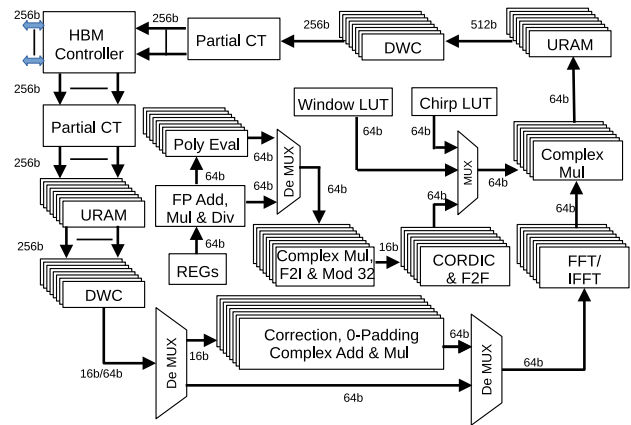
investigated fixed point onboard SAR focusing on FPGA. The current work aims to exploit the EO-ALERT embedded system experiences to data center processing utilizing an FPGA accelerator card. The goal of highly accurate phase preserving SAR images which are suited for a wide range of applications, is the motivation to introduce floating point arithmetics.

The hardware used for the EO-ALERT project was a prototyping board socketed with a Xilinx Zynq UltraScale+ ZU19EG multiprocessor system-on-a-chip (MPSoC). With adapted processing algorithm and distributing the workload between programmable logic (PL) and processing system (PS), the achieved processing times were approximately 3.7 s for image focusing in case of Multilook, Slant-range-Detected (MSD) or 4.3 s in regard to Single-look Slant-range Complex (SSC) per scene of about 375 km<sup>2</sup>. The FPGA design, proposed in this paper, is applicable as well for ground station processing as for on-board processing at an air-borne or space-borne sensor. Fig. 1 provides a basic overview of SAR focusing at ground station by off-loading it from CPUs to an accelerating card.

### 3 Overview of hardware

#### 3.1 Alveo data center card

AMD data center cards particularly the Alveo U55C offers a compelling solution for accelerating data workloads. It delivers high computing performance, flexibility, and energy efficiency. This enables data centers to achieve higher levels of computational power and can accelerate critical applications. The Alveo U55C accelerator card features a custom-built powerful Virtex XCU55 UltraScale+ FPGA from AMD and packed in with 16 GB of High-Bandwidth Memory generation 2 (HBM2), PCI Express (PCIe), and dual QSFP28 Ethernet ports capable of 100 Gbit/s each. The Alveo U55C data center accelerator card shown in the Fig. 1 is a single slot, full height, half length form factor passively-cooled device. Fig. 2 depicts the block diagram of U55C and its interfaces. Xilinx HBM2 is a memory technology that offers low-latency and high-performance data access for FPGA-based systems. HBM2 is designed to address the growing demand for memory bandwidth in applications such as AI, high-performance computing, and data centers. It is integrated on the same silicon pack-



**Figure 3** Data path of monochromatic  $\omega$ KA on Xilinx Virtex XCU55 UltraScale+ FPGA

age of FPGA and offers a memory bandwidth of up to 460 GB/s. Compared to traditional memory solutions like Double Data Rate generation 4 (DDR4), HBM2 delivers 20X more bandwidth but consumes more power. The Advanced eXtensible Interface (AXI) is a widely-used standard for connecting logic blocks in FPGA designs. It provides a high-bandwidth, low-latency, and scalable interconnect infrastructure. The integration of AXI enables an efficient and seamless communication between the FPGA fabric and the HBM2 memory. The 16 GB HBM2 is internally divided into 32 512 MB Pseudo Channels (PCs) for Alveo U55C accelerator card. The proposed SAR processing design efficiently connects the PCs through the 32 available AXI master interfaces into the HBM2 subsystem. The PCIe slot is the primary power source for the card, which provides a minimum of 75 W power. An additional 6-pin PCIe AUX power cable can deliver up to 150 W of power which allows increased performance and stability.

#### 3.2 Host server

A Dell PowerEdge Server R7525 hosts the Alveo U55C card in one of its PCIe slots. The server R7525 is configured to support up to three Alveo U55C cards, making it suitable for AI, ML, and virtualization workloads. Such a setup would allow that three cards may run up to three image focusing tasks in parallel or to setup a pipeline consisting of image focusing, post processing and AI inference. Although, it is not necessary to have a strong server to host an Alveo card, but running the development tools and FPGA design simulations on the same host, the Dell PowerEdge R7525 provides sufficient computing resources. The AMD tools Vitis and Vivado are used to develop and simulate the design and to generate the final bitstream.

### 4 SAR focusing

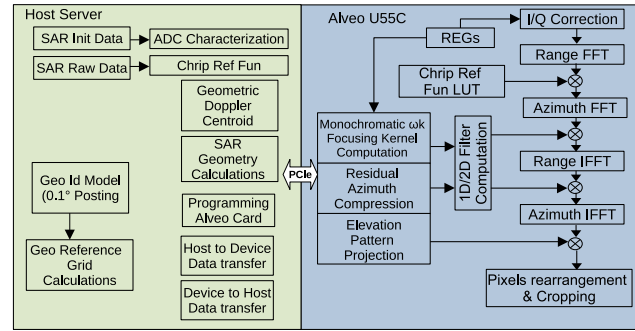
The most common spectral domain SAR focusing algorithms are Range-Doppler Algorithm (RDA), the Chirp Scaling Algorithm (CSA), and the wavenumber domain  $\omega$ K Algorithm ( $\omega$ KA). The  $\omega$ KA [7] is an advanced SAR

focusing technique that addresses the limitations of RDA and is less complex than CSA. An even less complex variant is the monochromatic  $\omega$ K algorithm, which approximates the range variant range migration sufficiently accurate for smaller swathes and medium resolution images. In contrast to the original  $\omega$ KA, the monochromatic variant omits the cumbersome implementation of Stolt mapping. The algorithm flow of monochromatic  $\omega$ KA on U55C accelerator card is shown in Fig. 4. The raw data is derived from TerraSAR-X single polarization StripMap (SM) mode data. This mode has a swath width of 30 km and provides a resolution of approximately 2 m in both dimensions, range and azimuth. In addition to the raw data, respective radar instrument settings as well as attitude and orbit data derived from the output of the satellite's attitude and orbit control system (AOCS) supplement the input raw data. The block size for SM processing is set to accommodate precisely 8192 azimuth lines and a maximum of 32,768 range pixels, which are represented as 8-bit/8-bit complex integer values. Depending on the actual acquisition parameters such a block covers an area ranging from 375 to 500 km<sup>2</sup>. The focused image generated on U55C is a SSC represented in 32-bit/32-bit complex floating point values. In order to store and process SAR raw data, intermediate data and the focused image, a minimum of 12.5 GB of memory is allocated in the HBM. In which, 0.5 GB is for raw data from first eight PCs, 8 GB is for intermediate data from next sixteen PCs, and lastly, 4 GB for focused data from the last eight available PCs. The focused image is accompanied with a geo reference grid computed on the host server, to map the image radar time coordinates to geographical coordinates.

#### 4.1 Software implementation

The host software (SW) running on the server calculates all geometry-related parameters depending on the acquisition-specific orbit state vectors and attitude quaternions and then transfers the parameters to registers (REGs) on the FPGA. The SW uses orbit interpolations and Zero-Doppler iterations to derive the acquisition geometry of the data take at a few supporting points and to determine the geographic coordinates of the final image. Based on the acquisition geometry, polynomials are fitted which compactly represent the filter coefficients of the matched filters used in the  $\omega$ KA. The replica of the transmitted chirp pulse is converted into complex reciprocal of its spectrum and then transferred to HBM.

The host SW is also responsible for the programming of the Alveo card, starting the hardware application which is also called an accelerating kernel and data management between the host and the card. The PCIe interface is used for communication between the host and card to transfer the data. The device's HBM serves as a global memory, accessible by both host and hardware accelerator applications. In order to exploit the parallel PCs on the HW, the raw data is rearranged into eight parts already on the host server. Each part contains every eighth azimuth line and is transferred to one of the first eight PCs of HBM. This gives an advantage to access and process eight lines in parallel on



**Figure 4** Functional block diagram of monochromatic  $\omega$ KA of host SW and hardware kernel on Alveo U55C accelerator card

the hardware. After focusing, the data is present in the last eight PCs of HBM in interleaved order. The host accesses it and then rearranges the data.

#### 4.2 Hardware implementation

The signal processing chain of monochromatic  $\omega$ KA is an accelerating kernel implemented on Alveo U55C. Though, a floating point design is costly with respect to resources and performance, the large FPGA of U55C allows to implement the design in single-precision floating-point. The main components of the signal processing chain are the in phase and quadrature phase (I/Q) raw data correction, FFTs and IFFTs in both range and azimuth, matched filter parameters generation and multiplications of filter kernels with the output of FFT data.

The matched filter parameters are calculated in parallel to the FFTs. Thus, they do not cause any extra clock cycles in the overall processing time. The applied filters are:

- spectral hamming window in range
- matched filter based on azimuth and range frequency-dependent terms of  $\omega$ KA in the 2-D spectral domain
- matched filter based on azimuth frequency-dependent and range dependent terms  $\omega$ KA in the range-Doppler domain
- window function consisting of the reciprocal azimuth antenna pattern of the sensor
- spectral hamming window in azimuth
- window function consisting of the reciprocal elevation antenna pattern of the sensor projected onto the focused image

Computation of these filters requires different mathematical operations as they are additions, multiplications, divisions, polynomial evaluations, complex exponentials, and type conversions. In the EO-ALERT project, these operations were performed in SW and then transferred as look-up tables to FPGA [4]. But here, in order to save latency and resources required to store these values on the FPGA, we compute them as a part of the datapath at the accelerating kernel.

At the hardware, the same FPGA resources are used for all polynomial evaluations. Thus, the hardware design for polynomial evaluation is driven by the maximum polynomial degree occurring in the algorithm. This is the window function consisting of the reciprocal range antenna pattern, where a polynomial of degree 6 is used. For the other polynomials, the unused coefficients of higher order terms are set to 0. A Horner's method as following (1) lowers the number of arithmetic operations to perform

$$f(x) = (((((a_6x + a_5)x + a_4) \\ x + a_3)x + a_2)x + a_1)x + a_0 \quad (1)$$

where  $a$ 's are the polynomial coefficients. The polynomial is evaluated in floating point using several Xilinx Multiply and Accumulator Intellectual Property (IP) cores. It takes 26 clock cycles to evaluate the polynomial when an input is given and it also supports streaming inputs. Thus, provision of polynomial outputs requires careful synchronization within the datapath.

#### 4.2.1 Datapath

A Finite State Machine (FSM) is implemented on the FPGA to control the data flow. Once SW initiated the kernel, the FSM takes over control. Firstly, it fills the replica of the transmitted chirp pulse from HBM to a Block RAM (BRAM) and then fills the raw data in the Ultra RAMs (URAMs). Employing the available BRAM resources on the Virtex XCU55 UltraScale+ FPGA, a maximum of eight parallel datapath instances can be instantiated. The main blocks of the datapath have been realized using IP cores from Xilinx, such as FFT, RAM, Coordinate Rotation Digital Computer (CORDIC), Data Width Converters (DWCs), and floating-point math operators. Before and after FFTs, a cache made up of eight parallel URAMs are placed to store data on chip. The implemented on chip caching also serves as an efficient corner turning (CT) approach. CT is an important step in SAR focusing to solve the issue of the data being unaligned after each signal processing step. The available size of URAM in the FPGA allows for  $32 \times 32$  k range lines or  $128 \times 8$  k azimuth lines to be cached at the beginning and end of the datapath at a time. Because of this limit, sub-blocks of SAR data are buffered repeatedly from HBM to URAMs. Next, the buffered data goes through bit width conversion and then gets streamed to the FFTs. Each output value coming out of the FFTs is multiplied with one filter coefficient generated on the FPGA and the results are written to the URAMs. Each datapath instance shown in Fig. 3 processes  $4 \times 32$  k range lines or  $16 \times 8$  k azimuth lines in one go. Combining all eight datapaths,  $32 \times 32$  k range lines or  $128 \times 8$  k azimuth lines processed all together. In total,  $2 \times 8192 \times 32$  k ranges lines and  $2 \times 32768 \times 8$  k azimuth lines are processed. Running eight parallel datapaths each containing one FFT IP core significantly decreases the overall latency since the FFTs contribute the dominant part to the latency. In the final step after azimuth IFFT, the data in each PC of the HBM is unaligned so the data is rearranged using URAMs. In this process also unnecessary pixels are removed.

## 5 Results

### 5.1 Resource utilization

Table 1 shows the resources utilized to develop SAR focusing algorithm on the Alveo U55C card. As it can be seen, the BRAMs and URAMs are the most critical resources as they are required in the FFT and in the caching of the data on chip.

**Table 1** Resource utilization of the SAR focusing

Site Type	Used	Available	Utilization[%]
Slice Registers	556K	2607K	21.3
Slice LUTs	501K	1303K	38.5
Block RAM Tiles	1818	2016	90.2
UltraRAM Tiles	608	960	63.3
DSP Slices	1358	9024	15.1

### 5.2 Latency

The primary objective of utilizing the Alveo U55C data acceleration card was to enhance the speed of SAR focusing. It is worth noting that programming an FPGA initially takes approximately 10 s for the U55C card, along with the allocation of buffers on the host for data synchronization with HBM. This requires 0.25 s. However, once this setup process is completed, SAR focusing can be efficiently performed on multiple azimuth blocks, each encompassing 256 megapixels.

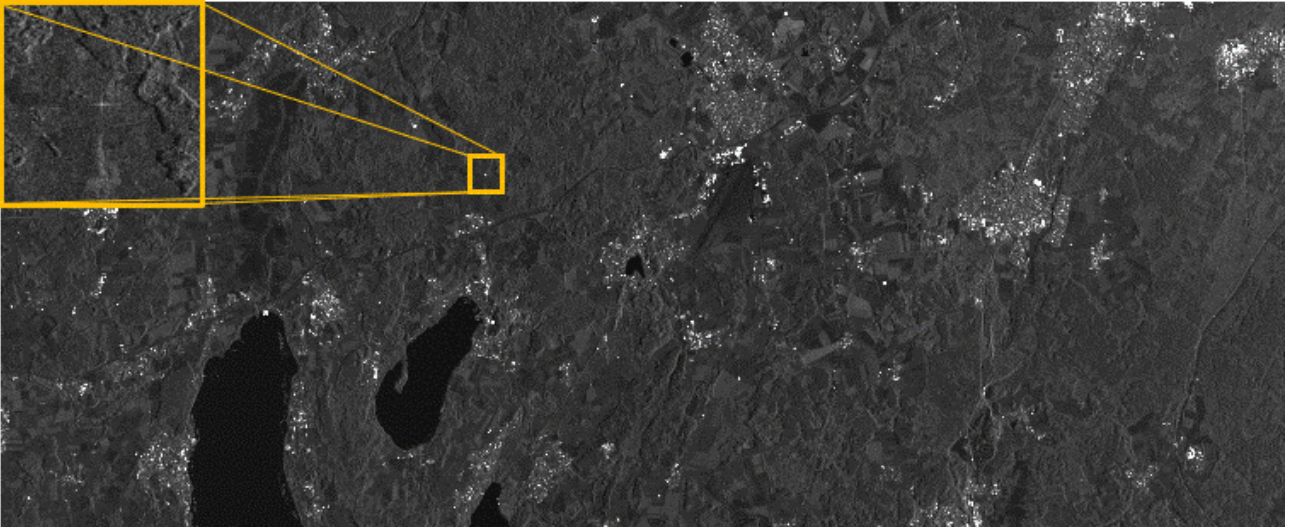
**Table 2** Processing time of SAR image generation

Action	Time[s]	Percentage[%]
Host to card data transfer	0.10	5.12
SAR focusing kernel	1.48	75.90
Card to Host data transfer	0.37	18.98
Total	1.95	100

Table 2 shows the data transfer times between host and card and as well processing time of the SAR focusing kernel. The maximum clock rate at which the kernel can run is 130 MHz. The time required for focusing one azimuth block is 1.48 s. For comparison, the sensor acquired raw data within 2.21 s. Thus, the focusing time is faster than the acquisition time which enables real-time focusing.

### 5.3 Quality analysis

The quality of the focused image was analyzed in detail on a SAR acquisition test site with corner reflectors to get the parameters of the impulse response function. The selected test site is located at Oberpfaffenhofen, Germany, equipped with four corner reflectors. SAR SM raw data was acquired by TerraSAR-X satellite and focused by the operational TerraSAR-X processor. By means of reverse SAR processing, the test raw data for FPGA processing was recon-



**Figure 5** Focused image of Oberpfaffenhofen site with four corner reflectors processed on Alveo U55C accelerator card. One of the corner reflectors is shown in the close-up. TerraSAR-X data © DLR 2017

**Table 3** Results of point target analysis on corner reflectors at the DLR test site Oberpfaffenhofen, Germany. The measured point target parameters from the TerraSAR-X image and from the Alveo U55C image are compared against each other. The measured values for the four individual corner reflectors are averaged.

Parameter	TerraSAR-X SSC	Alveo U55C SSC	Difference SSCs	Unit
Azimuth 3 dB width	1.564	1.566	$0.003 \pm 0.0003$	m
Range 3 dB width	1.3003	1.3003	$0.0 \pm 0.0002$	m
Peak power (cal)	47.62	47.52	$-0.11 \pm 0.04$	dB
Peak phase	*	*	$-0.38 \pm 0.02$	deg
Peak azimuth position	*	*	$-0.0005 \pm 0.0$	m
Peak range position	*	*	$0.0003 \pm 0.0$	m
Azimuth PSLR	-30.68	-30.56	$0.12 \pm 0.04$	dB
Range PSLR	-30.30	-30.32	$-0.02 \pm 0.08$	dB
2-D ISLR	-16.36	-16.53	$-0.17 \pm 0.15$	dB
Clutter power (cal)	-8.27	-7.96	$0.31 \pm 0.23$	dB
Signal energy (cal)	55.22	55.12	$-0.10 \pm 0.04$	dB m <sup>2</sup>
Main lobe energy (cal)	55.12	55.02	$-0.10 \pm 0.04$	dB m <sup>2</sup>

(\*=averaging of this parameter does not lead to a meaningful value)

structured from the SSC data since the TerraSAR-X mission does not provide level 0 raw data products. A point target analysis was performed on the basis of the four corner reflectors for both images, the original TerraSAR-X SSC and the FPGA processed SSC. The processing bandwidths of the test scene are 2.765 kHz in azimuth and 100 MHz in range for both processors. The pulse repetition frequency amounts to 3.716 kHz and the range sampling frequency is 110 MHz. The zero-Doppler velocity is 7053 m/s.

Figure 5 shows the image generated by the accelerating card. Table 3 lists and compares the measured parameters of the impulse response functions. It can be seen that the results of the operational TerraSAR-X processor can be reproduced by the U55C implementation. The average difference in azimuth and range position is at sub millimeter level while the standard deviation of these differences is

below measurement accuracy of the point target analysis. As the clutter power is slightly increased, the signal energy is decreased. The differences in signal to clutter ratio are mainly caused by the introduction of noise because multiple additional signal processing steps are applied in reverse SAR processing and FPGA focusing.

## 6 Conclusion

The presented work demonstrates that the SAR focusing developed in floating point can be done in real time. This is achieved by off-loading the compute intensive tasks from a CPU to an accelerating card equipped with an FPGA.  $8192 \times 32768$  pixels of SAR raw data are focused in 1.48 s. As the host server is able to accommodate up to three accel-

erating cards, a longer raw data acquisition can be split into  $3 \times 256$  megapixels and processed in parallel. The comparison of point target responses reveals that the FPGA SAR focusing kernel is able to generate highly accurate SSC images.

## 7 Literature

- [1] G. Li, F. Zhang, L. Ma, W. Hu, and W. Li, "Accelerating sar imaging using vector extension on multi-core simd cpu," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 537–540, 2015.
- [2] T. Jin, H. Wang, and H. Liu, "Design of a flexible high-performance real-time sar signal processing system," in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pp. 513–517, 2016.
- [3] T. Yang, Q. Xu, F. Meng, and S. Zhang, "Distributed real-time image processing of formation flying sar based on embedded gpus," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 6495–6505, 2022.
- [4] S. Wiehle, S. Mandapati, D. Günzel, H. Breit, and U. Balss, "Synthetic aperture radar image formation and processing on an mpsoc," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.
- [5] Y. Cao, S. Jiang, S. Guo, W. Ling, X. Zhou, and Z. Yu, "Real-time sar imaging based on reconfigurable computing," *IEEE Access*, vol. 9, pp. 93684–93690, 2021.
- [6] B. Li, H. Shi, L. Chen, W. Yu, C. Yang, Y. Xie, M. Bian, Q. Zhang, and L. Pang, "Real-time spaceborne synthetic aperture radar float-point imaging system using optimized mapping methodology and a multi-node parallel accelerating technique," *Sensors*, vol. 18, no. 3, 2018.
- [7] R. Bamler, "A comparison of range-doppler and wavenumber domain sar focusing algorithms," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, no. 4, pp. 706–713, 1992.