

# SAR SATELLITE ON-BOARD SHIP, WIND, AND SEA STATE DETECTION

*Stefan Wiehle, Dominik Günzel, Björn Tings*

German Aerospace Center (DLR)  
Maritime Safety and Security Lab  
Am Fallturm 9, 28359 Bremen, Germany

## ABSTRACT

This paper describes a prototype implementation of ship, wind, and sea state detection algorithms for satellite on-board SAR processing designed for Maritime Situation Awareness. Existing algorithms were adapted to run on a Multi-Processor-System-On-Chip (MPSoC) combining an FPGA and an ARM CPU and further optimized for fast runtime on the system. The achieved processing times were 20 s for ship detection and 16 s for sea state detection on a 29 Mpx SAR image. SAR processing is one component of a larger prototype system being developed in the frame of the H2020 project EO-ALERT, which further comprises an optical data chain, data compression/encryption, and delivery on multiple MPSoC boards.

*Index Terms*— SAR, on-board processing, ship detection, sea state detection

## 1. INTRODUCTION

A growing number of Earth Observation satellites is brought into orbit, many of them utilizing SAR sensors for their day-and-night and all-weather capabilities. The normal processing chain for satellite imagery consists of acquisition, satellite flight time to the nearest ground station, data downlink, image generation and processing at the ground station, and transfer to the user. The flight time to the ground station can be saved by bringing the image generation, processing and delivery onto the satellite. The product can be directly made available online with satellite-to-satellite communications and downloaded by the user. This short reaction time is especially important for Maritime Situation Awareness: the retrieved information, such as ship positions, wind speeds and wave heights, are very time sensitive and become deprecated within minutes.

In the EO-ALERT project [1], a prototype on-board system is built and tested supporting both optical and SAR measurements. The processing chain hardware consists of five boards, each accommodating a Multi-Processor-System-On-Chip (MPSoC) device. The boards will be individually configured for the different tasks such as optical processing, SAR

processing, data compression and encryption [2], and downlink. The results of the processing are downlinked as so-called alerts, small packages of data up to 10 kB, each containing the retrieved information on one ship or on sea state in a single area. For the whole system from payload data availability to reception of the alerts by the end-user, a total latency of 5 min is required in the project.

This paper focuses on the adaptation and implementation of the SAR ship and sea state detection algorithms.

## 2. HARDWARE AND SYSTEM OVERVIEW

Each board in the EO-ALERT prototype features a Xilinx Zynq UltraScale+ ZU19EG MPSoC, which provides programmable logic (PL) and a quad-core processing system (PS) based on ARM Cortex-A53 CPUs running a Linux-based operating system (OS). Communication and data exchange between both components are provided via the Advanced eXtensible Interface (AXI). The PS and PL are each equipped with 4 GiB of DDR4-SDRAM for storage of processing data and, in the case of the PS, to run the operating system.

The on-board SAR chain includes the sequential steps SAR level 1 (L1) and level 2 (L2) processing. L1 processing constitutes generation of the focused SAR image from raw data delivered by the SAR instrument as shown in [3]. Since the EO-ALERT project does not include design and development of such an instrument, TerraSAR-X single polarization StripMap mode data is used. The images produced by the on-board L1 processing have a size of up to  $3500 \times 12000$  pixels with coverage of  $375 \text{ km}^2$ , whereby pixels represent the backscatter coefficient  $\sigma_0$  scaled to 16 bit unsigned integer. The focused images including their respective ancillary data serve as input for the L2 processing. Within the scope of the EO-ALERT project, SAR L2 processing involves generation of products with either ship detection or sea state detection information, and will be explained in the following sections.

Prior to implementation of the SAR processing chain on the target MPSoC, the utilized L1 and L2 algorithms were analyzed to decide upon a suitable hardware/software partitioning. This is a crucial part of the design phase to ensure

**Table 1:** Runtimes of ship detection algorithm on Intel Core i5-6440HQ with base clock of 2.6 GHz and 1.2 GHz ARMv8 Cortex-A53 with and without dedicated hardware CFAR core. Test scene used was  $3488 \times 8320$  pixels, with 750 m CFAR background and 375 m guard window (99 px and 50 px, respectively).

	Initial det.	Refinement	Ambig. filter	Land rem.	Total
x86	66.2 s	1.1 s	3.7 s	0.8 s	73.9 s
ARM	731.6 s	8.3 s	4.1 s	2.2 s	748.0 s
PS/PL	3.8 s	8.3 s	4.1 s	2.2 s	20.3 s

that the available board resources are allocated to the SAR L1 and L2 processing steps with regard to maximizing the overall performance for fastest possible product delivery. At the same time, the higher complexity of a dedicated hardware implementation compared to a software approach has to be considered. The initial analysis showed that the full L1 and L2 SAR chain could be implemented on a single ZU19EG MPSoC while achieving high processing performance. However, caching mechanisms for L1 image focusing required all of the available UltraRAM and most of the Block RAM resources on the ZU19EG [3], calling for a memory-efficient implementation of the L2 processing.

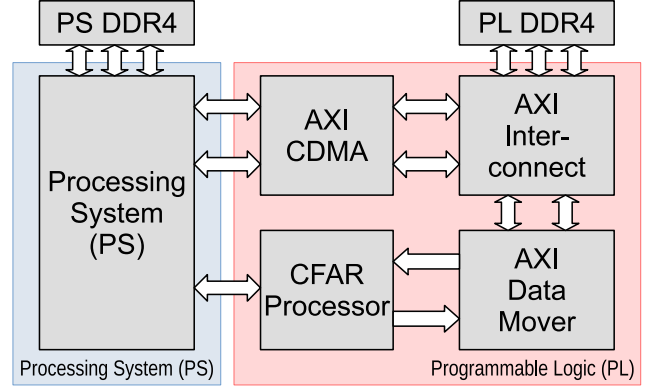
### 3. SHIP DETECTION

The ship detection algorithm is adapted from [4] and [5]. The essential steps are initial detection, refinement, ambiguities filtering, and land removal.

Software unit testing of all processing steps was performed on the ARM processing system. Table 1 lists the computation times for the steps when executed on different CPU architectures. A SAR image with  $3488 \times 8320$  pixels and pixel spacing of roughly 3.8 m was used. As expected, the quad-core ARM PS was overall significantly slower than a quad-core x86 CPU.

Most notably, the initial detection took eleven times as long to compute on the PS. During this step potential ship candidates are identified by application of a Constant False Alarm Rate (CFAR) algorithm, where the brightness of each pixel is compared to the statistics of the background area around it. Because these statistics have to be computed for every pixel of the image, CFAR has high computational complexity. At 12.2 min, this first step alone would exceed the target system latency of 3.5 min by far, and it was therefore decided to implement the CFAR algorithm in the PL in order to exploit the potential speed-up of a dedicated hardware processor.

The core presented in [6] served as the basis for the CFAR processor in this project. It is able to process the CFAR statis-



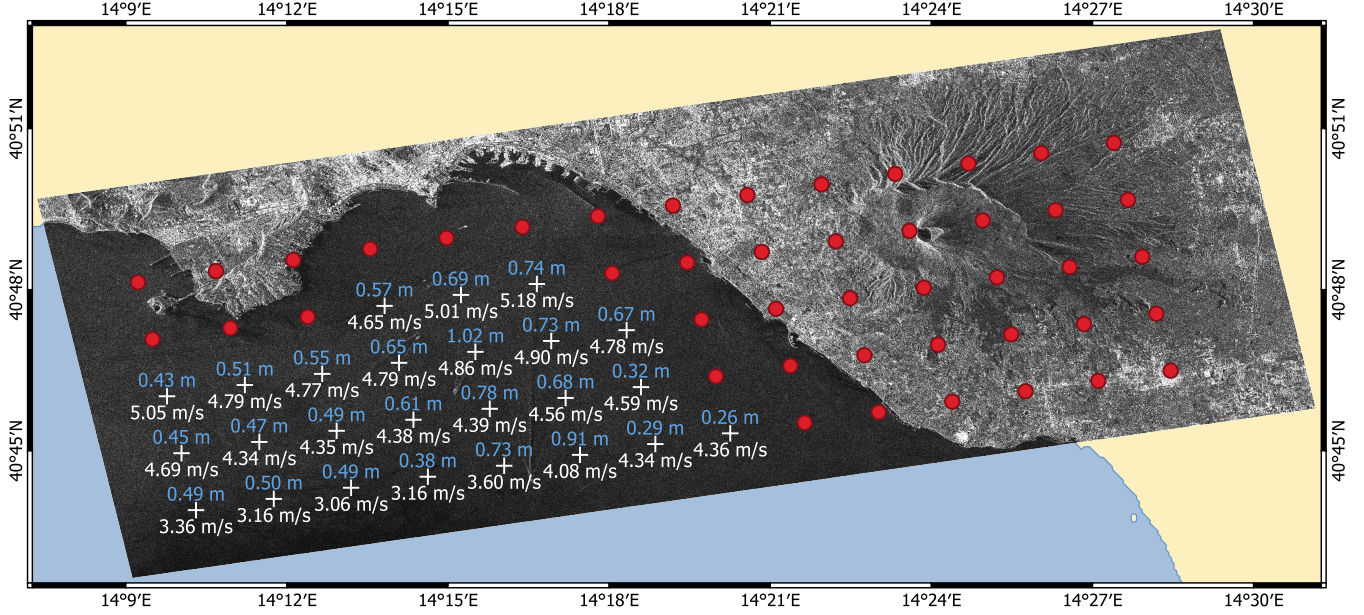
**Fig. 1:** CFAR processor integration into Zynq US+ ZU19EG.

tics of one pixel every five clock cycles by utilization of a highly pipelined data path. Modifications have been made for ship detection and to adapt the core to the target hardware.

Furthermore, the processor has been extended to support changing of the CFAR background and guard window dimensions in pixels at run-time, which were previously hard-coded and could only be changed by synthesizing the entire core again. While the background and guard windows sizes are set to 750 m and 375 m, respectively, for every scene – the same values as used operationally at DLR ground station Neustrelitz since many years – the pixel spacing of the images produced by the on-board L1 processing [3] varies from 3 m to 5 m depending on the acquisition geometry and the pixel sizes of the resulting windows vary accordingly. Run-time configurability of these parameters therefore helps to keep detection rates consistent across different acquisitions.

The diagram in Fig. 1 shows the integration of the CFAR core into the Zynq UltraScale+ ZU19EG and the general datapaths in the MPSoC. Connected to the PS is a Xilinx AXI CDMA (Central Direct Memory Connect), which the PS uses for high-throughput data transfers between its own DDR4-SDRAM and the PL-side DDR4-SDRAM. An AXI Interconnect in front of the PL-DDR allows additional components to access it, in this case an AXI DataMover, which is controlled by the CFAR processor and used to transfer image data from and results back to the PL-DDR. The DataMover converts between the memory-mapped AXI domain and AXI-Stream domain, which the CFAR processor expects to continuously fill its internal Block RAM cache. The SAR images generated by L1 processing are stored in the PL-side DDR4-SDRAM, hence it has been defined as common data storage between the L1 and L2 steps as well as the PS.

When a SAR image is ready to be processed and stored in the PL-DDR, ship detection is started by execution of a program on the PS. First, the image and its metadata are transferred to the PS-DDR via the CDMA, and the parameters for the CFAR algorithm are calculated in software. These parameters are written by the PS to registers inside the CFAR core



**Fig. 2:** Results of wind speed detection (white numbers) and sea state detection (blue numbers) demonstrated for a scene at Naples, Italy. White crosses indicate centers of processed grid cells, red dots mark centers of cells not processed due to land filtering. TerraSAR-X data © DLR 2016.

**Table 2:** Resource utilization of the implemented CFAR detection circuit on the Zynq UltraScale+ ZU19EG FPGA.

Site Type	Used	Available	Utilization
Slice Registers	28K	1045K	2.7 %
Slice LUTs	41K	523K	7.8 %
Block RAM Tiles	145	984	14.7 %
DSP Slices	260	1968	13.2 %

and then the CFAR processing is initiated. While processing, the binary results (whether pixels have been detected as a ship candidate or not) are collected by the CFAR core in an internal Block RAM queue and written in bursts to the PL-DDR via the DataMover. Once the image has been fully processed, the CFAR processor triggers an interrupt for the PS, which then transfers the results from the PL-side into its own DDR4 via the CDMA for the subsequent detection steps.

The hardware design has been synthesized and implemented on the Zynq UltraScale+ ZU19EG with a target frequency of 125 MHz for all blocks. Table 2 shows the FPGA resource utilization of the L2 ship detection circuit. Even after the outlined extensions to the CFAR core the required resources are low compared to what is available on the ZU19EG MPSoC. The need for a memory-efficient design has been fulfilled because just 13.2% of the available Block RAM and none of the UltraRAM is used.

The dedicated CFAR core in the PL processed the same SAR image with  $3488 \times 8320$  pixels and otherwise equal pa-

rameters in 2.7 s. Including the time required for transfer of the CFAR results from the PL-side DDR4-SDRAM to the PS, the latency of the initial detection step was 3.8 s (Tab. 1). In total, it took 20.3 s for the full ship detection, which is well within the latency budget.

#### 4. WIND SPEED AND SEA STATE DETECTION

The wind speed and sea state are calculated on a grid with cell sizes of about  $2 \times 2 \text{ km}^2$ . Land masking is applied before the calculations, so all grid cells containing land can be skipped.

Wind speed processing uses the XMOD2 geophysical model function (GMF) to calculate the wind speed from the measured sea surface backscatter [7]. It is computationally very fast as it works with the averaged intensity of all pixels in a grid cell.

Sea state detection analyses several spectral and GLCM (gray level co-occurrence matrix) parameters, as well as the previously calculated wind speed, to determine the wave height [8, 9]. Several performance improvements were made to the code, some of them general optimizations to reduce the calculation complexity (e.g. removing unnecessary output), others specifically to increase performance on the ARM architecture. For example, repeated loops over each pixel in the whole image proved to have significantly higher latency impact on the ARM PS than on x86 beyond difference in clock speeds. Since ARM is a load-store architecture, manipulation of variables in system memory requires more CPU cycles compared to the register-memory architecture of x86,

**Table 3:** Runtimes of wind and sea state detection algorithm on Intel Core i5-6440HQ x86-CPU and 1.2 GHz ARMv8 Cortex-A53 CPU both before and after optimization. The test scene used (shown in Figure 2) was  $3488 \times 8320$  pixels, with 25 valid measuring grid cells out of 70 total.

	Land removal	Wind	Sea state	Total
x86	1.1 s	0.1 s	3.3 s	5.4 s
ARM	8.2 s	0.4 s	38.4 s	49.0 s
ARM opt.	2.5 s	0.4 s	12.4 s	16.4 s

which allows direct manipulation of variables in memory. Therefore, accesses to all pixels of the image were combined in as few loops as possible.

Table 3 shows the achieved processing times of an x86 CPU and of the ARM PS before and after optimization. These runtimes are below the latency limit and allow running the processing only on the PS without implementing it on the PL. Figure 2 shows exemplary results for a scene with wind and sea state detection processed on the ZU19EG.

## 5. CONCLUSIONS

This paper shows that performing ship detection as well as wind and sea state detection on board of a satellite is possible within very reasonable latency requirements. The CFAR algorithm applied for ship detection had to be implemented in hardware to achieve the required runtimes. This was not necessary for the wind and sea state detection, where optimizations of the code enabled the algorithms to run only on the ARM processor of the MPSoC system.

These results enable further work towards a new generation of Earth Observation satellites with similar processing capabilities as used here available on-board, providing users with products less than 5 min after acquisition.

## 6. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 776311. The EO-ALERT project is coordinated by DEIMOS Space. More information on the project at eo-alert-h2020.eu.

## 7. REFERENCES

[1] M. Kerr, A. Fiengo, J.-I. Bravo, R. Hinz, F. Membibre, A. Latorre, H. Breit, S. Wiehle, D. Günzel, S. Mandapati, U. Balss, O. Koudelka, F. Teschl, E. Magli, T. Bianchi, M. Martina, R. Freddi, M. Benetti, G. Curci, F. Milani, and C. Marcos, “EO-ALERT: A Novel Architecture for

the Next Generation of Remote Sensing Satellites Supporting Rapid Civil Alerts,” in *IAC 2020*, October 2020.

- [2] M. Caon, P.M. Ros, M. Martina, T. Bianchi, E. Magli, F. Membibre, A. Ramos, A. Latorre, M. Kerr, S. Wiehle, H. Breit, D. Günzel, S. Mandapati, U. Balss, and B. Tings, “Very Low Latency Architecture for Earth Observation Satellite Onboard Data Handling, Compression, and Encryption,” in *2021 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2021)*, Brussels, Belgium, July 11-16, 2021 (subm.).
- [3] H. Breit, S. Mandapati, and U. Balss, “An FPGA/MPSoC based Low Latency Onboard SAR Processor,” in *2021 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2021)*, Brussels, Belgium, July 11-16, 2021 (subm.).
- [4] S. Bruschi, S. Lehner, T. Fritz, M. Soccorsi, A. Soloviev, and B. van Schie, “Ship Surveillance With TerraSAR-X,” *IEEE T. Geosci. Remote*, vol. 49, no. 3, pp. 1092–1103, 2011.
- [5] Björn Tings, Carlos Augusto Bentes da Silva, and Susanne Lehner, “Dynamically adapted ship parameter estimation using TerraSAR-X images,” *Int. J. Remote Sens.*, vol. 37, no. 9, pp. 1990–2015, 2016.
- [6] D. Gregorek, D. Günzel, J. Rust, S. Paul, D. Velotto, J. Imber, B. Tings, and A. Frost, “FPGA Prototyping of a High-Resolution TerraSAR-X Image Processor for Iceberg Detection,” in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, 2019, pp. 1832–1835.
- [7] X.-M. Li and S. Lehner, “Algorithm for Sea Surface Wind Retrieval From TerraSAR-X and TanDEM-X Data,” *IEEE T. Geosci. Remote*, vol. 52, pp. 2928–2939, May 2014.
- [8] A. Pleskachevsky, W. Rosenthal, and S. Lehner, “Meteorological Parameters for Highly Variable Environment in Coastal Regions from Satellite Radar Images,” *ISPRS J Photogramm.*, vol. 119, pp. 464 – 484, 2016.
- [9] A. Pleskachevsky, B. Tings, and S. Jacobsen, “Sea State from Sentinel-1 SAR Wave Mode Imagery for Maritime Situation Awareness,” in *EUSAR 2020*, Mar 29-Apr 01, 2021.