Very Low Latency Architecture for Earth Observation Satellite Onboard Data Handling, Compression, and Encryption

VERY LOW LATENCY ARCHITECTURE FOR EARTH OBSERVATION SATELLITE ONBOARD DATA HANDLING, COMPRESSION, AND ENCRYPTION

Michele Caon, Paolo Motto Ros, Maurizio Martina, Tiziano Bianchi, Enrico Magli[⋆], *Francisco Membibre, Alexis Ramos, Antonio Latorre, Murray Kerr*[†], *Stefan Wiehle, Helko Breit, Dominik Günzel, Srikanth Mandapati, Ulrich Balss and Björn Tings*[‡]

[⋆]Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy
[†]Deimos Space S.L.U, Madrid, Spain
[‡]German Aerospace Center (DLR), Cologne, Germany

ABSTRACT

In modern society, the ever-increasing demand for Earth Observation products in a large variety of sectors is exposing the limitations of traditional satellite data chain architectures. The European Union Horizon 2020 EO-ALERT project aims at overcoming the existing bottlenecks by leveraging the performance of state-of-the-art commercial off-the-shelf devices to move the critical elements of data processing on the flight segment without sacrificing processing performance. This paper introduces the architecture of the EO-ALERT CPU Scheduling, Compression, Encryption and Data Handling Subsystem, responsible for coordinating the onboard optical and Synthetic Aperture Radar data chains, as well as providing data compression, encryption, and storage services. The performance obtained by a reference implementation of the proposed architecture is also presented, showing an extremely low contribution to the overall system latency that allows real-time Earth Observation product delivery to the end user in less than $5\,\mathrm{min}$.

Index Terms— Earth Observation, Onboard Compression and Encryption, Satellite Architecture, Low Latency, Real-Time.

1. INTRODUCTION

Traditionally, the generation of Earth Observation (EO) products has been achieved by compressing the acquired raw data onboard and transmitting it to the ground segment, where all the image generation and processing tasks are executed. Data transmission usually represents the most significant bottleneck in this process, resulting in a Near Real-Time (NRT) provision of the products to the end user, typically within $1\,\mathrm{h}$ to $3\,\mathrm{h}$ [1] from acquisition time (of which $10\,\mathrm{min}$ to $20\,\mathrm{min}$ after downlink). The EO-ALERT project aims at achieving true real-time EO product delivery. Specifically, the system is optimized to deliver rapid civil alerts to the end user in less than $5\,\mathrm{min}$ while supporting both optical (single- and multi-band) and Synthetic Aperture Radar (SAR) data. To this purpose, commercial off-the-shelf (COTS) components are employed together with space-grade ones to move the key processing tasks on the flight segment [2], therefore eliminating most of the data transfers to the ground. Figure 1 shows the top-level physical architecture of the EO-ALERT avionics system. The Processing Unit, which encloses all the scheduling, data handling, and processing functions, is composed of up to seven Multiprocessor System on a Chip (MPSoC) devices, each featuring a multi-processor ARM-based Processing System (PS) as well as Programmable Logic (PL) resources to host both software (SW) applications and high-performance hardware (HW) accelerators. Redundant configurations can be employed to increase radiation and fault tolerance depending on the target mission requirements. The resulting system is a scalable and highly reconfigurable platform suitable for several EO missions in a wide range of applications, capable of handling computationally expensive tasks such as image generation and processing.

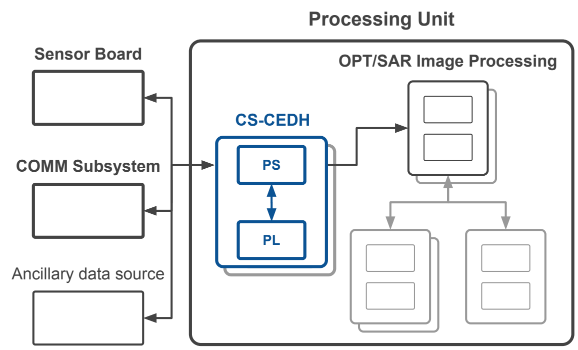

Fig. 1: EO-ALERT top-level physical architecture.

This paper describes the architecture of the CPU Scheduling, Compression, Encryption and Data Handling (CS-CEDH) Subsystem. In a typical EO acquisition scenario, it performs the following actions: 1. Acquire the input raw and auxiliary

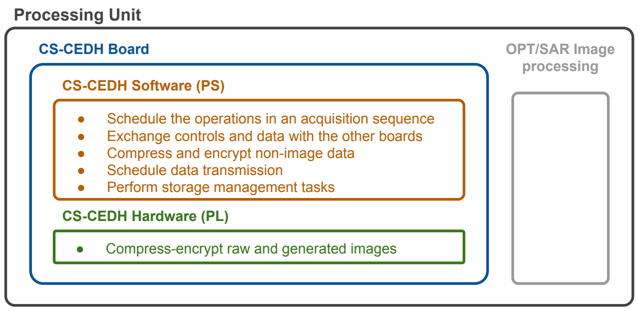

Fig. 2: CS-CEDH HW-SW partitioning.

data from the sensor(s) and other onboard sources. 2. Move the acquired input data to the Image Processing Boards. 3. Compress and encrypt the input data and store it on the onboard storage device. 4. Acquire the EO products from the Image Processing Boards. 5. Compress and encrypt the EO products and store them on the onboard storage device. 6. Forward the compressed and encrypted products to the Communications Subsystem for transmission to the ground segment. The CS-CEDH Subsystem schedules and executes these steps so that high-priority EO products (referred to as *alerts*) are delivered to the end-user device as soon as possible (*fast* data chain). Low-priority data (e.g., raw sensor data) is transmitted to the ground segment only for diagnosis or verification purposes (*slow* data chain).

2. CS-CEDH HW-SW PARTITIONING

The CS-CEDH Board is inserted as a central node in a complex system whose functions and timing may vary depending on the mission requirements and target scenario. All its functions shall be supported by a combination of SW and HW, offering high-performance data processing and intelligent management of the onboard resources and interfaces. Compression and encryption tasks imply computationally expensive operations on a large amount of data, while all the other tasks (control functions and data transfers) can be classified as sparse and short workloads. Therefore, the PL resources of the CS-CEDH MPSoC are dedicated to accelerating the compression and encryption of image data, while all the other tasks are implemented inside the software running on the general-purpose multi-core Central Processing Unit (CPU) available in the PS, as shown in Fig. 2. The PL also hosts the interfaces driving the high data rate PCI links used to exchange data with the other boards inside the Processing Unit. The remaining interfaces are mapped to the PS resources. Namely, the CS-CEDH Subsystem is connected to the Communications Subsystem and the Sensor Board via Gigabit Ethernet interfaces, the onboard ancillary data source is accessed using a Universal Asynchronous Receiver-Transmitter (UART) interface, and the image processing boards can exchange control and status information with the CS-CEDH on a shared CAN bus. All the SW and HW components share the same physical memory to minimize the number of data transfers.

3. CS-CEDH IMPLEMENTATION AND MOTIVATIONS

To emulate the target avionics system [2], the CS-CEDH Subsystem is deployed on a Xilinx® Zynq® Ultrascale+ ZU19EG MPSoC. The image data compression HW accelerator is synthesized through High-Level Synthesis (HLS) starting from an optimized implementation of a compression and encryption algorithm [3] written in the C language and based on the Consultative Committee for Space Data Systems (CCSDS) 123.0-B-2 recommended standard [4, 5]. The resulting IP core maintains most of its software counterpart's functionalities, including (but not limited to) the possibility to configure at run time the compression level (lossless and near-lossless). Both single- and multi-band optical raw and generated images are supported, as well as complex SAR data. The accelerator communicates with the CS-CEDH PS via a high-performance AXI-4 interface. The completion of a compression-encryption task is communicated using a SW interrupt to avoid busy waiting.

Due to the system's complexity, the CS-CEDH SW is implemented as a heavily multi-threaded application software running on top of a light-weight yet complete GNU/Linux Operating System (OS), hosted by the feature-rich quad-core general-purpose ARM® Cortex® A53 microprocessor available in the PS of the CS-CEDH MPSoC. The multi-threaded SW architecture enables an asynchronous execution and coordination of all the tasks involved in the entire onboard data chain, eliminating idle times and ensuring real-time responsiveness in processing the requests and data from the surrounding Subsystems.

The scheduling algorithm of the CS-CEDH SW is carefully tuned to achieve the lowest possible latency in delivering high-priority data (i.e., alerts) to the Communications Subsystem and thus to the end-user device. To this purpose, all those tasks entering the critical path for the fast data chain are granted the highest execution priority and privileged access to shared HW and SW resources.

Synchronization with the Image Processing Subsystem is achieved using a handshake protocol based on *request* and *accept* commands exchanged over the shared CAN interface. Using this mechanism, the CS-CEDH Subsystem can issue several transfer requests to the target Image Processing Board, that is allowed to accept them in any order based on the availability of its computational resources. Similar commands are used to communicate with the mission control system and send or receive status and configuration parameters to change Processing Unit's behaviour at run-time. For instance, the CS-CEDH Board can be instructed to change the data compression level or request the removal of unnecessary data from the onboard storage device.

The data that must be delivered to the ground segment is forwarded by the CS-CEDH Subsystem to the Communications Subsystem using the dedicated Gigabit Ethernet link. The minimum transmission unit corresponds to a CCSDS Space Packet [6], tagged with relevant information about the transported data so that the most appropriate communication link can be selected for delivery to the ground segment (i.e., a very low-latency S-band transmitter or a high throughput Ka-band or optical transmitter [2]). The packetization of the compressed and encrypted data is performed just-in-time before its transmission to avoid wasting CPU and storage resources. A two-priority queue was implemented to grant to the encrypted alerts privileged access to the link, possibly stalling on-going transmissions of lower priority data.

SW compression and encryption services for non-image data (geo-referencing data, calibration parameters, floating-point projections, etc.) are provided through general-purpose SW compression and encryption routines. Compression was implemented using the *deflate* core [7] from the well-known Zlib general-purpose compression C library, which offers a reasonable compromise in terms of (selectable) compression efficiency and throughput, as well as compatibility with virtually any device. Encryption is performed by a SW stream cypher that uses a Pseudo-Random Generator (PRG) based on an assembly-optimized implementation of the resource-efficient Keccak cryptographic sponge functions [8].

To reduce the impact of data loss during the transmission of compressed and encrypted data to the ground segment, the CS-CEDH SW splits the largest image data into several tiles, each containing a configurable number of lines. Each image tile is individually compressed and encrypted by the HW accelerator and individually reconstructed on the ground segment. If some data is lost during the transmission, only the affected tiles will result in corrupted samples in the recovered image data. While smaller tiles can further reduce the penalty of transmission errors, their cost in terms of compression efficiency can significantly impact the transmission latency. In the reference implementation, the number of lines per tile was set to 128 to keep the overhead on the overall compression ratio within 3% from that obtained compressing the entire image at once.

4. RESULTS

The implementation of the CS-CEDH Subsystem described in the previous sections was tested by connecting it to a PC running accurate emulators of all the surrounding Subsystems. All the PCI data transfers were replaced by I/O operations on the onboard storage device, while handshake and control commands were exchanged as TCP/IP packets on top of a dedicated Gigabit Ethernet link. At first, all the system features were comprehensively verified in compliance with the project's requirements. Then, performance-oriented tests using real-world input data were performed. The following paragraphs report the preliminary results obtained during such tests.

Resource utilization analysis reported, for the entire CS-CEDH Subsystem, 75% of the combinational logic blocks and less than 40% of the sequential elements available in the PL of the MPSoC. The worst-case memory usage was measured to be $4.4\,\mathrm{GB}$ when acquiring and buffering twenty $50\,\mathrm{Mpixel}$, $16\,\mathrm{bit}$ optical raw data and taking into account HW-dedicated buffers, leaving up to $3.6\,\mathrm{GB}$ available for other avionics functions or buffering of additional input data.

The overall image compression-encryption throughput achieved by the hardware accelerator is $13.8\,\mathrm{MiB/s}$ to $19.4\,\mathrm{MiB/s}$ depending on the type of data and the selected compression level. The SW compression throughput for non-image data ranges from $4\,\mathrm{MiB/s}$ to $15\,\mathrm{MiB/s}$ based on the data content and obtainable level of SW parallelism. Finally, the SW encryption routines achieve similar performance, and in particular, the average throughput of alert encryption was measured to be $(10.6 \pm 1.0)\,\mathrm{MiB/s}$ across the entire set of alerts ($\sim$$10\,\mathrm{KiB}$ to $20\,\mathrm{KiB}$ each). Regarding data transmission, the throughput of on-the-flight packetization proved to be compatible with the bandwidth of the Gigabit Ethernet link, enabling a transmission throughput of $800\,\mathrm{Mbit/s}$ to $950\,\mathrm{Mbit/s}$.

The three plots in Fig. 3 show the scheduling and execution time of the SW compression-encryption, HW compression-encryption tasks, and data transmission tasks, respectively, as executed by the CS-CEDH Subsystem during four simulations covering all the foreseen employment scenarios: ship detection using two single-band $50\,\mathrm{Mpixel}$ optical data ($0\,\mathrm{s}$ to $40\,\mathrm{s}$); extreme weather detection using 5-band $0.7\,\mathrm{Mpixel}$ plus $7\,\mathrm{Mpixel}$ High-Resolution Visible (HRV) optical data ($40\,\mathrm{s}$ to $60\,\mathrm{s}$); ship and extreme weather detection using $200\,\mathrm{Mpixel}$ SAR data ($60\,\mathrm{s}$ to $120\,\mathrm{s}$ and $120\,\mathrm{s}$ to $200\,\mathrm{s}$). Each task is reported as a short segment representing its execution time. The vertically stacked tasks in the first and third plots correspond to encryption and transmission tasks on the generated alerts in each case (26, 1, 13, and 40 alerts, respectively). As shown, these tasks are scheduled as soon as possible and completed with very low latency. The oblique sets of tasks in the second plot correspond to the (serial) compression and encryption of raw and generated image data tiles on the HW accelerator.

Thanks to the results outlined in the previous paragraphs, the CS-CEDH Subsystem achieves extremely low latencies both in the fast and slow data chains, as shown in Table 1 in case of lossless compression (worst-case). Because raw data compression-encryption is performed in parallel to image generation and processing, which are more time-consuming tasks, the associated latency does not enter the critical path for the slow data chain. This allows the EO-ALERT system to deliver high-priority compressed and encrypted EO prod-

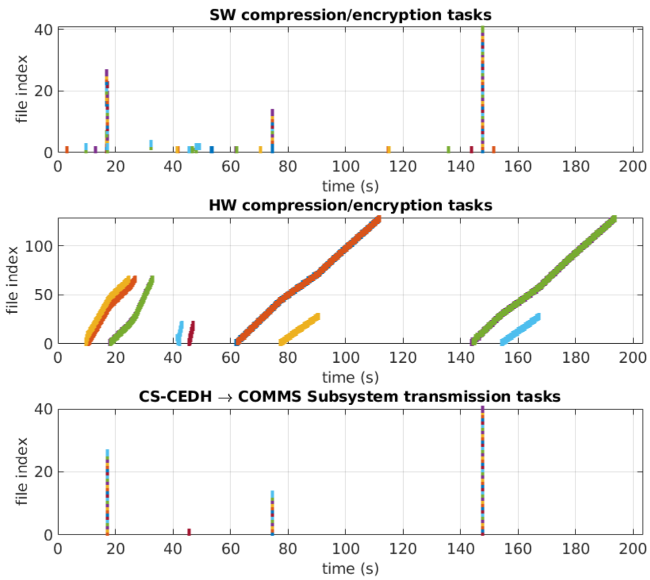

Fig. 3: Scheduling of processing and transmission tasks on the CS-CEDH Subsystem with optical and SAR data.

Optical single-band	Pre-processing [ms]	Processing [s]	TX [ms]
raw data	0.34 ± 0.03	10.0 ± 0.1	449 ± 30
gen. image	135 ± 3	10.0 ± 0.1	273 ± 19
alerts (100×)	0.74 ± 0.09	0.096 ± 0.008	7.56 ± 0.14

Optical 5-band	Pre-processing [ms]	Processing [s]	TX [ms]
raw data	0.13 ± 0.05	1.474 ± 0.001	145 ± 11
gen. image	31 ± 6	2.411 ± 0.008	176 ± 10
alerts (100×)	1.13 ± 0.12	0.18 ± 0.02	19.1 ± 2.3

SAR	Pre-processing [ms]	Processing [s]	TX [ms]
raw data	0.33 ± 0.01	50.7 ± 2.6	2306 ± 45
gen. image	281 ± 17	3.99 ± 0.10	483 ± 12
alerts (100×)	0.72 ± 0.07	0.078 ± 0.011	9.30 ± 0.17

Table 1: CS-CEDH latency for image data and alerts.

ucts to the end user in under 5 min from the acquisition of the raw data in all the foreseen application scenarios and within 1 min in specific cases. Most importantly, these results show a contribution of the CS-CEDH Subsystem to the fast data chain (i.e., alert delivery) of just 200 ms when dealing with up to 100 generated alerts. Even under heavy load conditions, this latency is maintained under 1 s.

5. CONCLUSIONS

This paper proposed a novel architecture for onboard EO data acquisition and processing that leverages state-of-the-art high-performance COTS device to achieve true real-time EO product delivery to the end user. In particular, it was shown that a carefully tuned scheduling of onboard operations, supported by heavily optimized processing elements, results in extremely low latencies when compared to traditional, ground-based systems.

6. ACKNOWLEDGEMENTS

This work and the EO-ALERT project have been funded and supported by the European Union Horizon 2020 research and innovation programme under grant agreement No 776311.

7. REFERENCES

[1] D. K. Davies, M. E. Brown, K. J. Murphy, et al., “Workshop on using NASA data for time-sensitive applications [space agencies],” *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 3, pp. 52–58, 2017.

[2] M. Kerr, S. Tonetti, S. Cornara, et al., “EO-ALERT: A novel architecture for the next generation of earth observation satellites supporting rapid civil alerts,” in *71st International Astronautical Congress (IAC)*, 2020.

[3] D. Valsesia and E. Magli, “High-throughput onboard hyperspectral image compression with ground-based CNN reconstruction,” *IEEE transactions on geoscience and remote sensing*, vol. 57, no. 12, pp. 9544–9553, 2019.

[4] The Consultative Committee for Space Data Systems (CCSDS), *Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression*, vol. Blue Book, CCSDS Secretariat, February 2019, Recommended Standard CCSDS 123.0-B-2.

[5] A. Kiely, M. Klimesh, I. Blanes, et al., “The new CCSDS standard for low-complexity lossless and near-lossless multispectral and hyperspectral image compression,” in *ESA On-Board Payload Data Compression Workshop, OBPDC*, September 2018.

[6] The Consultative Committee for Space Data Systems (CCSDS), *Space Packet Protocol*, vol. Blue Book, CCSDS Secretariat, September 2003, Recommended Standard CCSDS 133.0-B-1.

[7] P. Deutsch, “RFC1951: Deflate compressed data format specification version 1.3,” 1996.

[8] A. Migliorati, T. Bianchi, and E. Magli, “Selective encryption in the CCSDS standard for lossless and near-lossless multispectral and hyperspectral image compression,” in *Image and Signal Processing for Remote Sensing XXVI*, Lorenzo Bruzzone, Francesca Bovolo, and Emanuele Santi, Eds. International Society for Optics and Photonics, 2020, vol. 11533, pp. 221 – 228, SPIE.