Robust Quantum Communication for Space Systems Through an Heterogeneous RISC-V based FPGA/SoC Platform

Robust Quantum Communication for Space Systems Through an Heterogeneous RISC-V based FPGA/SoC Platform

Giorgio Cora
Politecnico di Torino
Turin, Italy
giorgio.cora@polito.it

Arash Amini Bardpareh
Politecnico di Torino
Turin, Italy
arash.aminibardpareh@polito.it

Gonzalo Miguel Joaquin Fernandez Lobo
Università di Padova
Padova, Italy
gonzalomigueljoaquin.fernandezlobo@unipd.it

Corrado De Sio
Politecnico di Torino
Turin, Italy
corrado.desio@polito.it

Sarah Azimi
Politecnico di Torino
Turin, Italy
sarah.azimi@polito.it

Andrea Stanco
Università di Padova
Padova, Italy
andrea.stanco@unipd.it

Luca Sterpone
Politecnico di Torino
Turin, Italy
luca.sterpone@polito.it

Abstract

Quantum Key Distribution (QKD) is increasingly adopted in security-critical links and future scenarios will include also space applications where radiation-induced faults can compromise the correctness and availability of the protocol. This risk is amplified on SRAM-based FPGAs, where configuration upsets can alter circuit behavior. Through this paper, we present a robust, heterogeneous FPGA/SoC platform to validate the reliability of QKD sifting on commercial off-the-shelf hardware. A lightweight RISC-V processor supervise the entire procedure, while the sifting accelerator is coupled with a lightweight monitoring unit to detects faults and triggers error correction through partial reconfiguration. The platform has been implemented on AMD ZCU102 UltraScale+ development boards and evaluated through fault injections. The adopted mitigation techniques allows downtime reduction by about 12000× and lowers the sifting-module error rate by around 2%.

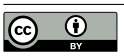
CCS Concepts

• **Computer systems organization → Reliability**.

Keywords

FPGA, QKD, Reliability

ACM Reference Format:
Giorgio Cora, Arash Amini Bardpareh, Gonzalo Miguel Joaquin Fernandez Lobo, Corrado De Sio, Sarah Azimi, Andrea Stanco, and Luca Sterpone. 2026. Robust Quantum Communication for Space Systems Through an Heterogeneous RISC-V based FPGA/SoC Platform. In *Proceedings of the 23rd ACM International Conference on Computing Frontiers (CF '26), May 19–21, 2026, Catania, Italy*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3801487.3801825



CF '26, Catania, Italy

ACM ISBN 979-8-4007-2568-5/26/05
https://doi.org/10.1145/3801487.3801825

1 Introduction

Quantum Key Distribution (QKD) enables two remote parties, conventionally Alice and Bob, to establish secure keys by exploiting fundamental properties of quantum communication rather than computational hardness assumptions [7]. In QKD, information is shared through quantum states, called *qubits*, and measured at the receiver. Any attempt to extract information from the transmitted states inevitably introduces disturbances that can be detected statistically. This security model makes QKD attractive for long-term confidentiality, especially in scenarios where advances in computing can weaken conventional cryptographic protocols.

Among the various application domains, satellite-assisted QKD is emerging as a practical approach to extend key distribution beyond the distance and infrastructure constraints of ground optical links [14], enabling connectivity between geographically distant ground nodes. Space-oriented platforms, however, face specific constraints in terms of size, weight, power, and operational availability [13]. This motivates heterogeneous FPGA/SoC solutions, where compute-intensive post-processing stages are accelerated in reconfigurable logic, which combines high throughput and deterministic streaming behavior with flexible integration into a system-level control plane. At the same time, the adoption of SRAM-based FPGAs raises reliability concerns in radiation-prone environments. Single-event effects can corrupt configuration memory and persistently alter circuit behavior until repaired, potentially affecting both service continuity and the correctness/consistency of the generated keys.

In this work, we propose a heterogeneous FPGA/SoC platform that implements the sifting stage in programmable logic and emulates a basic two-device Alice–Bob QKD communication scheme. An ideal quantum-channel with no losses and no errors is emulated over a high-speed optical serial link managed by a lightweight

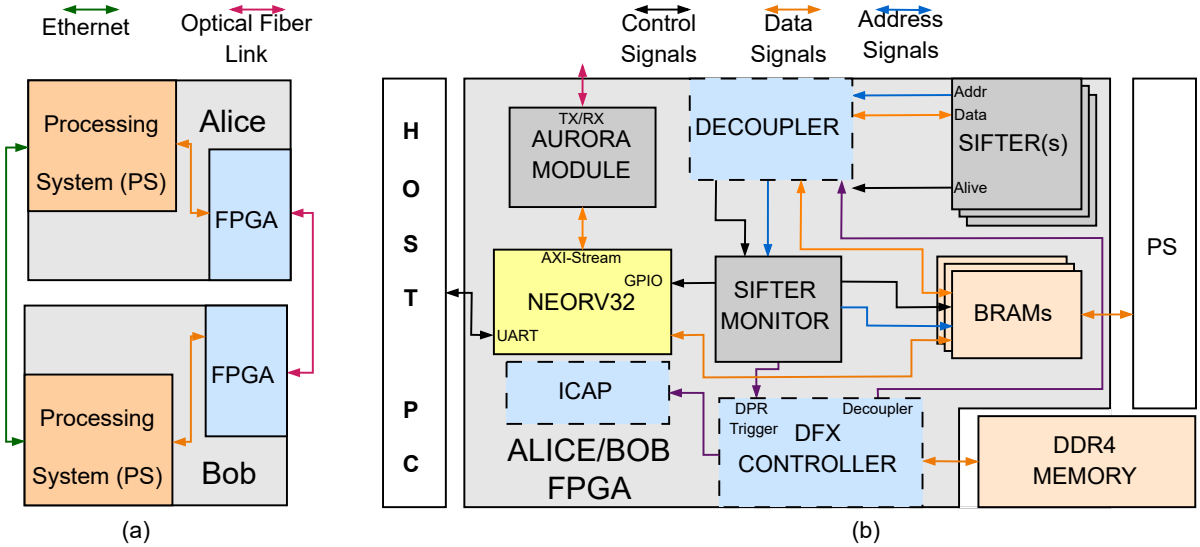


Figure 1: Overview of the implemented Alice–Bob platform (a) and internal FPGA datapath used to implemented the sifting procedure(b)

RISC-V core, while the classical channel and system control are handled by the processing system via Ethernet. We integrate lightweight monitoring and recovery mechanisms to supervise the sifting module and improve robustness under radiation-like faults. The platform is evaluated through a fault-injection campaign targeting FPGA configuration memory, enabling a quantitative assessment of failure modes, recovery behavior, and their impact on sifting correctness and execution continuity. The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed architecture and its fault-handling mechanisms. Section 4 reports the experimental methodology and results from the fault-injection campaign. Finally, section 5 concludes the paper.

2 State of the Art

QKD is attracting growing interest for space-oriented applications, as it can overcome optical-fiber distance constraints by leveraging free-space optical links between satellites and ground stations [10]. After quantum transmission and detection, the parties must execute a classical post-processing pipeline including sifting, parameter estimation, and key-distillation steps such as information reconciliation (IR) and privacy amplification (PA) [12].

As post-processing throughput requirements become more tight, these stages can become practical bottlenecks, motivating heterogeneous implementations where compute-intensive kernels are offloaded to specialized hardware. FPGA/SoC platforms are therefore increasingly explored as accelerators for QKD post-processing thanks to reconfigurability, deterministic streaming behavior, and high performance [5]. For example, [8] presents a reconfigurable multi-protocol key-distillation framework on FPGA using HLS to address throughput and interoperability. Similarly, FPGA-based implementations of reconciliation and other post-processing building blocks have been demonstrated, positioning reconfigurable logic as an enabling technology for sustaining high-rate IR/PA pipelines[11].

For space deployments, however, performance must be complemented by resilience to radiation effects. SRAM-based FPGAs are susceptible to Single Event Effects (SEEs), and configuration-memory upsets may persist until repaired, potentially altering circuit behavior or disrupting operation. Several studies, including [9], have analyzed how SEEs and Multiple Bit Upsets (MBUs) affect FPGA configuration memory and how error accumulation can lead to functional failures. Mitigation techniques such as scrubbing/readback-based approaches[4] and partial (dynamic) reconfiguration to reduce repair time and scope have been investigated over the years. In parallel, fault-injection tools targeting FPGA configuration memory have been developed to emulate radiation effects and evaluate detection and recovery behavior[6], enabling controlled assessment of radiation-induced faults in reconfigurable devices.

Despite the growing adoption of FPGA-based acceleration for QKD post-processing and the extensive literature on radiation effects and mitigation in reconfigurable devices, limited analyses have been provided on the impact of radiation-induced faults on QKD-oriented systems, particularly in terms of end-to-end sifting correctness and execution continuity. This work addresses this gap by proposing a scalable architecture for QKD-oriented sifting, complemented by lightweight supervision mechanisms to improve robustness under radiation-induced faults.

3 Proposed Architecture

Fig. 1 shows the overall Alice and Bob structure. The platform is organized around three main functional units. A lightweight control unit based on a RISC-V processor manages the whole platform, including data movement and execution control, a hardware sifting structure and its supervisory logic have been implemented to accelerate the post-processing stage and provide a fault detection mechanism to improve the overall system's reliability and finally, a repair mechanism based on Dynamic Partial Reconfiguration (DPR) restores corrupted sifting logic in case of errors, limiting overall downtime and key generation consistency.

In space-oriented QKD, the quantum channel can be implemented through free-space optical links, with qubits encoded in photons and measured at the receiver. However, reproducing photon generation and free-space transmission would require dedicated hardware, significantly increasing experimental complexity. Therefore, to maintain the essential dual-channel structure of QKD while reducing setup complexity, we emulated the quantum channel by transmitting the raw data stream over an optical-fiber Aurora link between the two boards. The classical channel is implemented through TCP/IP communication over Ethernet using the on-board PS. This partition also reflects realistic deployments, where accelerated post-processing blocks are managed by a general-purpose processing element responsible for control, synchronization, and system-level fault handling.

3.1 NEORV32 Processor

The execution of the sifting procedure, together with the Aurora-based optical-fiber link used to emulate quantum data delivery, is managed by a dedicated soft core implemented in the PL, the NEORV32[15], an open-source RV32I processor written in VHDL and well suited for FPGA integration, with optional extensions that can be enabled depending on system needs. The core interfaces with the Aurora subsystem through an AXI-Stream datapath. Memory-mapped access to the on-chip BRAMs used to store bases, data bits, and sifting results is provided via AXI-Lite. UART is used to send status and logs to an external host PC. Finally, dedicated GPIO lines are used for controlling the sifters and receiving interrupts and status signals from the monitoring logic.

3.2 Sifting module and monitoring controller

The sifting operation is implemented as a dedicated hardware module to sustain high input rates and reduce post-processing latency. To improve robustness in the presence of SEE that may affect the logic, the sifting unit is paired with a lightweight controller responsible for runtime monitoring and recovery actions.

The controller monitors execution in two ways. First, the sifting core exposes a dedicated signal that toggles once per iteration. This mechanism is effective to detect stalls due to deadlocks or halt in the system. Additionally, the controller also monitors the memory access pattern by tracking the read/write address signals issued by the sifting core. By observing address evolution, the controller can detect anomalous behaviors such as repeated accesses to the same locations, out-of-range sequences, or inconsistent read/write progression. Upon detecting a fault condition, the controller asserts an interrupt to the NEORV32 for event logging and, in parallel, triggers the error correction procedure through Dynamic Partial Reconfiguration (DPR) using AMD's Dynamic Function eXchange (DFX)[3] flow managed by the DFX Controller [1]. In contrast to full device reconfiguration, DPR restores only the reconfigurable region hosting the failing sifting module by reloading a golden partial bitstream, while the rest of the FPGA logic remains operational. This localized recovery enables fast restoration of the faulty unit, reducing overall downtime without the need of stopping unrelated modules. Finally, to support execution resumption after repair, the controller stores the last valid execution state required to restart the computation (e.g., current address or iteration index). Once DPR completes, the sifting core resumes operation from the stored state.

The platform is scalable through module replication: multiple independent sifting modules can be instantiated to increase throughput, while a single monitoring unit can supervise multiple instances, keeping additional control overhead minimal.

4 Experimental Results

The platform was implemented on two AMD UltraScale+ ZCU102 boards. Table 1 reports the resource utilization for a single board instance and includes one sifting module considering a standard implementation of the BB84 protocol with 2 mutually unbiased bases. When multiple sifting instances are integrated, utilization increases accordingly.

Table 1: Platform Resources Utilization on ZCU102

Platform Modules	LUTs	FFs	BRAMs	DSP
NEORV32	1445	1317	4	0
Sifter	313	418	0	0
Controller	56	295	0	0
DPR Logic	1185	989	0	0
Glue Logic	2654	4335	9	0
Total[%]	2.06%	1.34%	1.42%	0%

Overall utilization remains low, leaving margin to instantiate additional sifting modules, integrate further QKD post-processing modules, or include stronger mitigation techniques, such as applying TMR to the main processor. Glue logic includes resources related to system integration blocks, such as AXI interconnects, bus adapters, and memory controllers, and the Aurora IP. The current design operates at 100 MHz, currently limited by the interconnections between the processor and Aurora. Each sifting module compares Alice's and Bob's bases bit by bit, processing 32 bits in 32 clock cycles; when including the additional cycles required to fetch the next bases word and store the computed key, this corresponds to a throughput of approximately 95 Mcomparisons/s.

4.1 Fault injection procedure

To evaluate FPGA designs under radiation-induced effects, two approaches are commonly adopted: radiation testing, which reproduces both the fault source and its physical effects through particle irradiation, and fault emulation, which reproduces the effect by injecting bit-flips into configuration memory. While radiation tests provides the most realistic validation, it requires specialized facilities and are extremely costly. For this reason, fault emulation is widely used as an effective and scalable alternative to extensive experimental campaigns. In this work, SEEs in configuration memory are emulated using the AMD SEM-IP[2], which supports runtime injection of configuration-memory errors without repeatedly reprogramming the device with corrupted bitstreams. This is particularly beneficial for large campaigns, as the full configuration of the ZCU102 typically takes on the order of tens of seconds, making repeated full reconfiguration impractical. Injections are restricted to configuration-memory regions associated with the implemented design, representing the worst case scenario. The corresponding injection addresses are generated using pyxel[6]. We

conducted two complementary campaigns to assess the sensitivity of the sifting module and the behavior of the general platform under configuration upsets. In each campaign, 10,000 injections are performed by randomly selecting configuration bits within the target logic.

4.2 Fault injection results

Table 2 summarizes the results of the fault-injection campaign targeting the sifting module.

Table 2: 10'000 Fault Injection Results for the Sifting Module

Error Type	#Events	Occurrence%
No Error	9080	90.8%
Undetected SDC	738	7.38%
Detected SDC	182	1.82%
Total	10'000	100%

Detected SDC indicates runs in which the sifting module is affected by an error, but the monitoring unit detects the anomaly and triggers recovery via DPR, allowing execution to resume and complete correctly. Undetected SDC refers to runs that complete but produce an output that differs from the golden reference. These results should be interpreted as a worst-case scenario. Since the campaign targets configuration bits associated specifically with the sifting module, a higher error incidence is expected compared to a whole-device campaign that samples uniformly across all configuration bits. A second campaign targeting the full platform was also conducted. Table 3 reports the corresponding results.

Table 3: 10'000 Fault-injection results for the full platform

Outcome	#Events	Events [%]
No error	9305	93.05%
System halt	438	4.38%
Communication errors	188	1.88%
SDC	49	0.49%
Detected SDC	18	0.18%
Controller halt	2	0.02%
Total	10'000	100%

Faults are classified as follows: System halt, when the entire board stopped working; Communication errors, when inter-board data exchange contains corrupted data; Detected SDC, when the monitor detects an error and triggers recovery, leading to the correct final result; Controller halt, when the monitoring module halts; and SDC, when the sifting output differs from the golden reference. As expected, most faults lead to a system-level halt. When multiple sifting instances are implemented, however, a larger portion of the programmable logic would be dedicated to QKD-specific processing, and radiation-induced faults can more easily affect key generation.

Additionally, the platform was evaluated in terms of downtime and recovery latency. A full reconfiguration of the ZCU102 takes approximately 12s through JTAG, while it is reduced to about 5–6s when booting from SD. In contrast, Dynamic Partial Reconfiguration restores only the failing sifting module and completes in about 0.93ms, yielding a speedup of roughly 12.9k× and 5.4k–6.5k× compared to JTAG- and SD-based full reconfiguration, respectively. If multiple sifting modules are affected, they are reconfigured sequentially, but recovery still remains orders of magnitude faster than a full reconfiguration, significantly reducing overall system downtime. Finally, the bit error rate (BER) for the runs leading to SDC was computed. The average BER across SDC runs is 19.6% for injections targeting the sifting module only, while it decreases to 12.7% when targeting the full platform.

5 Conclusions

This paper presented a heterogeneous FPGA/SoC platform for space-oriented QKD post-processing, implementing sifting in programmable logic and integrating lightweight monitoring and recovery. Fault injection results demonstrate the effectiveness of partial reconfiguration as a repair mechanism and show how radiation-induced configuration upsets can affect both availability and correctness in SRAM-FPGA-based QKD post-processing, motivating mitigation and recovery mechanisms to improve overall reliability and preserve correct key generation in space-oriented scenarios.

Acknowledgments

This study was carried out within the "REASE - REsilient computing Architecture in the Space quantum communication Era" project, CUP number E53D23008300006 – funded by the Ministero dell'Università e della Ricerca – within the PRIN 2022 program (D.D.104 - 02/02/2022) and (partially) supported by the ICSC—Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union—NextGenerationEU. This manuscript reflects only the authors' views and opinions and the Ministry cannot be considered responsible for them.

References

[1] AMD, Dynamic Function eXchange Controller v1.0 Product Guide (PG374) 2025.
[2] AMD: Soft Error Mitigation Controller v4.1, (PG036) 2018.
[3] AMD, Vivado Design Suite User Guide: Dynamic Function eXchange (UG909) 2025.
[4] Fredrik Brosser et al. 2014. Assessing scrubbing techniques for Xilinx SRAM-based FPGAs in space applications. 296–299. doi:10.1109/FPT.2014.7082803
[5] Jeremy Constantin et al. 2017. An FPGA-Based 4 Mbps Secret Key Distillation Engine for Quantum Key Distribution Systems. *Journal of Signal Processing Systems* 86, 1 (01 Jan 2017), 1–15. doi:10.1007/s11265-015-1086-1
[6] Corrado De Sio et al. 2023. PyXEL: Exploring Bitstream Analysis to Assess and Enhance the Robustness of Designs on FPGAs. 1–4. doi:10.1109/SMACD58065.2023.10192116
[7] Charles H. Bennett et al. 2014. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science* 560 (2014), 7–11. doi:10.1016/j.tcs.2014.05.025 Theoretical Aspects of Quantum Cryptography – celebrating 30 years of BB84.
[8] Foram P Shingala et al. 2022. Real time QKD Post Processing based on Reconfigurable Hardware Acceleration. arXiv:2211.17019 [cs.CR] https://arxiv.org/abs/2211.17019
[9] Kyle W. Gear et al. 2022. An analysis of FPGA configuration memory SEU accumulation and a preventative scrubbing technique. *Microprocessors and Microsystems* 90 (2022), 104467. doi:10.1016/j.micpro.2022.104467
[10] Orsucci Davide et. al. 2025. Assessment of Practical Satellite Quantum Key Distribution Architectures for Current and Near-Future Missions. 43, 3 (2025). doi:10.1002/sat.1544
[11] Yang S. et al. 2023. Hardware design and implementation of high-speed multi-dimensional reconciliation sender module in continuous-variable quantum key distribution. *Quantum Inf Process* (2023). doi:10.1007/s11128-023-04124-9
[12] Chi-Hang Fred Fung et al. 2010. Practical issues in quantum-key-distribution postprocessing. *Phys. Rev. A* 81 (Jan 2010), 012318. Issue 1. doi:10.1103/PhysRevA.81.012318
[13] Yang Li et al. 2025. Microsatellite-based real-time quantum key distribution. *Nature* 640, 8057 (01 Apr 2025), 47–54. doi:10.1038/s41586-025-08739-z
[14] Sheng-Kai Liao et al. 2017. Satellite-to-ground quantum key distribution. *Nature* 549, 7670 (01 Sep 2017), 43–47. doi:10.1038/nature23655
[15] Nolting, S., et al.: The NEORV32 RISC-V Processor. Zenodo 2023. doi:10.5281/zenodo.8260609