

National and Kapodistrian University of Athens

Department of Informatics and Telecommunications

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών
— ΙΔΡΥΘΕΝ ΤΟ 1837 —

PhD PhD Thesis

LDPC encoding hardware architectures for on-board processing datachains

Dimitrios Theodoropoulos (Δ695)

Time frame: April 2016 - September 2021

Supervisor

Prof. Antonios Paschalis

Advisor

Prof. Dimitris Gizopoulos

Prof. Nektarios Kranitis

Contents

1	Introduction	2
1.1	Forward Error Correction schemes	2
1.2	On-board data processing	3
1.3	Space communication channels, systems & protocols	5
2	Background	6
2.1	FPGAs in space	6
2.2	Space-grade CPUs	8
2.3	Spacewire and spacefibre	10
2.4	Bit-level channel coding	13
2.5	Magnetic recording media coding	13
2.6	Packet-level coding	13
3	Review of the state-of-the-art	14
3.1	QC LDPC bit-level codes	14
3.2	Packet level erasure codes	20
4	Goal of Thesis	22

1 Introduction

1.1 Forward Error Correction schemes

Forward Error Correction (FEC) coding schemes are used extensively in almost every communication and data processing system, in order to increase the reliability of transmission and storage of data. This is especially important in space communications scenarios, due to the extremely stringent SNR requirements of deep-space links, or conversely, the high data rates and low latency necessary in near-earth satellite communication scenarios. At the same time, an efficient FEC scheme implementation in a realistic scenario has to balance contradicting requirements and offer a variety of trade-offs in terms of error correcting efficiency, encoding/decoding complexity, throughput, hardware resources utilization and power consumption.

The traditional approach which has been widely adopted since the dawn of digital communications implements countermeasures against errors caused by noise, distortion and interference, at the symbol level. We refer to this component of the communication system as channel coding [1]. Almost every modern communication standard and system includes a channel coding scheme, at least as an option. Well known schemes in this area include Reed-Solomon (RS) [2] and Turbo [3] codes. Another highly advantageous class of channel codes are the Low-Density Parity-Check (LDPC) codes, which are linear block codes, characterized by large block lengths and sparse parity-check matrices. Introduced by R.G. Gallager in 1960 [4], LDPC codes had in following years generally succumbed to oblivion, due to the current eras technology limitations, which could not allow their implementation at a reasonable cost. However, advances in VLSI technology, together with the application of efficient code design techniques have annihilated those barriers. Among the entire range of modern error correcting codes (ECC), they are currently the most promising approach towards the capacity limit described by Shannon [5]. This has established them as the optimal choice for FEC in modern applications.

The initial Gallager codes were random and although they exhibited excellent error-correcting capabilities, hardware implementation was challenging. In order to reduce implementation complexity and encoding/decoding speed, additional structure has been designed into the parity check matrices of all practical LDPC codes in modern applications, so that they consist of an array of juxtaposed cyclic sub matrices, named the circulants, which can be efficiently implemented. These structured codes are collectively referred to as Quasi-Cyclic (QC) LDPC codes. QC-LDPC codes have been adopted by many modern communication standards, such as IEEE 802.11, 802.16 and DVB-S2. A special class of structured LDPC codes, the protograph-based QC codes have recently received considerable research interest in many modern standards. QC LDPC codes have also been adopted by the Consultative Committee for Space Data Systems (CCSDS) as recommended standard for on-board channel coding in Near-Earth and Deep-Space communications [6].

A multitude of encoder architectures for QC-LDPC codes has been proposed

in the literature, while several products are available in the market. Most of the proposed architectures though are optimized for the specific codes adopted by corresponding standards, leveraging the specific properties of the particular code structure. Especially those demonstrating practical throughput in the range of multiple Gbps, they are either entirely not applicable or are not expected to scale with CCSDS codes, the parity-check matrix of which do not exhibit the required structure.

The error correcting capability of bit-level channel coding however is limited in high speed and deep fading scenarios, such as those encountered in modern earth-to-satellite and satellite-to-satellite laser links. Moreover, these environments are characterized by high latency and traditional automatic repeat request (ARQ) schemes are not applicable or practical, or the fading effect of the communication channel is so deep that bit-level channel codes cannot provide the required reliability, since even a single scintillation effect can span multiple entire transmitted codewords. In these cases, the erroneously received or completely missed codewords can be considered as erased symbols and the most suitable model for the communication channel is the block erasure channel. Error correction in this case takes place at a higher level of the communications protocol stack than bit-level channel coding (which is typically a function of the data link layer in OSI protocol stack). Erased symbols are entire packets of the underlying protocol.

A common approach for coding over block erasure channels is the combination of RS codes with interleaving. RS codes are maximum distance separable (MDS) codes: if (n, k) are the dimensions of a code, it can recover from the erasure of any $n - k$ or fewer symbols. Consequently, they can provide optimal error recovery capability. An interleaver is typically connected to the output of the RS encoder in order to protect against deep fading. Such coding schemes have been proposed in [7] and [8] for optical space communications. The limitation of RS codes, however, is high encoding complexity, which imposes the use of short block lengths. The polynomial arithmetic operations involved in encoding and decoding operations result in non-linear encoding/decoding complexity, even in the base case proposed in [9]. The use of RaptorQ codes has alternatively been proposed in [10].

Another promising approach is the use of packet-level LDPC erasure codes, according to which encoded symbols are entire blocks of information bits. Although these codes are not MDS, capacity approaching ensembles can perform very close to the Singleton bound [11] and encoder and decoder complexity can scale linearly with block length. Packet-level LDPC erasure coding has also been proposed in [12] for near-earth and deep space communications.

1.2 On-board data processing

The stringent requirements of aerospace applications in terms of reliability and power call for a different approach, when considering on-board data processing equipment. Commercial devices, targeting hugely larger market shares and lower time to mar-

ket, cannot obviously meet these requirements. Processors in space are required to withstand harsh environmental conditions, mainly due to radiation effects. In addition, the risk margin of the disruption of the mission needs to be significantly lower. To meet these ends, the space industry has established the notion of Technology Readiness Level, and the relevant guidelines for ECSS are provided in [13].

The degradation of the reliability of electronic systems manifests itself in two forms of errors in their operation. The most severe form refers to hard errors. These can happen as a consequence of the gradual or sudden degradation of the system caused by the accumulation or a surge of total ionizing dose (TID) or atomic displacement (Total Non Ionizing Dose-TNID or Displacement Damage-DD) [14]. Another kind of effects are transient phenomena which lead to so-called "soft errors" in the component's operation. When the error in the system is caused by the passage of a single particle, the event can be categorised as Single Event Effect (SEE). SEEs can lead to soft errors, for example Single Event Upsets [15], or hard, as is the case with Single Event Upsets (SEU) or Burnouts (SEB).

Depending on the type of the effect, various mitigation techniques are applied at various levels: from the physical layer, which refers to the semiconductor fabrication process up to the system level design. Devices employing these techniques are referred to as radiation tolerant, or radiation hardened devices. Radiation hardening aims to minimise the probability of radiation effect's occurrence in the first place, mostly by measures on the physical layer and their cost of radiation hardened can be significantly higher than that of their commercial counterparts. Radiation tolerance, on the other hand, assumes that radiation effects are bound to occur and aims at reducing the impact of radiation effects on the system's operation. ECC in the memories and buses is the fundamental radiation tolerance technique. A summary of mitigation techniques at various levels of design can be found in [16] and the references therein.

The topic of mitigation techniques is widely covered in the literature. Consequently, we limit our brief description to the following techniques, which are more relevant to this work: Triple Modular Redundancy (TMR) and memory scrubbing. In a basic TMR scheme, three redundant circuits perform the same task on the same data. A majority vote process at the system's output can mask a failure in one of the circuits. Obviously, the cost of this approach is that it requires triple resources. Memory scrubbing, as the name implies, is a method to increase the integrity of data stored in a memory system. It requires that a method of ECC has been applied to the data written in the memory. Its contents are periodically retrieved, any errors are detected and corrected with the ECC and the result is written back to the memory. The frequency of the memory scans needs to be balanced, so that single errors are not accumulated and the ECC fails.

Typical on-board data handling systems are built around a central processor (OBC-On Board Data Computer), which is mostly responsible for telecommand functions and the coordination of the rest of the platform subsystems: telecommuni-

cation, telemetry, mass memory subsystems, sensors, instruments, and payload processors. All these subsystems communicate through highly reliable communication links, typically MIL-STD-1553, or spacewire and spacefibre, which are described separately in Section 2.3. The Space Avionics Open Interface Architecture (SAVOIR) initiative is a move towards the standardization of space avionics and, among other products, it proposes a reference functional architecture reference model.

In the near future space computing technology is expected to converge more rapidly with terrestrial practices [17], so that, depending on the mission goals, the required balance between performance, resiliency and cost is met: as smaller payloads with a limited lifespan are becoming more popular, the requirements for space-qualified parts can be relaxed. The most extreme example of this scenario is the case with cubesats and nanosatellites in the "new space" emerging trend [18], the lifespan of which can be as small as a few days [19]. In this aspect, for non-mission critical functions and time-specific payloads, even the use of COTS equipment can be acceptable.

1.3 Space communication channels, systems & protocols

near-earth and deep space channel modelling. Then about erasure channel Fundamentals: Capacity, FEC basics, channel models

2 Background

2.1 FPGAs in space

Initially, FPGAs implemented only auxiliary tasks and glue logic in a spacecraft system. The basic telemetry and flight control tasks are handled by specialized CPUs, which is the topic of subsection of 2.2. However, FPGAs have recently gained increased popularity for aerospace applications, due to the increased processing power and size, weight, power, and cost (SWAP-C), when compared to CPUs and GPUs [20]. In addition, While continuing to support these functions, FPGAs are widely used for embedded computing in Space. For example, in [21], a Hyperspectral Image Compression implementation for CCSDS 123.0-B1 recommended standard is introduced, which is built on a COTS SoC. FPGAs are the de-facto solution for demanding processing acceleration applications, like deep-learning algorithms in neural networks.[22].

There only a limited number of Radiation Hardened By Design (RHBD) FPGAs in the market. The most important device families are manufactured by Microsemi and Xilinx [23]. A common feature of these families is that the configuration memory is SRAM, instead of flash, since the latter technology is susceptible to radiation effects [24], with an obvious impact on the cost. The products of both vendors share a rich mission heritage, an extensive overview of which is also provided in [23].

In addition to the standard hardening techniques, Microsemi PolarFire radiation tolerant FPGAs chips use Silicon-Oxide-Nitride-Silicon (SONOS) Non-Volatile (NV) technology [25], which provides immunity against SEU effects, in addition to low power. The physical layer manufacturing details of the SONOS technology, as well as its rad-hard attributes are widely covered in [25]. TMR in the user logic, when required, is assured through suitable provisions from the bundled software (Libero simplify).

On the other hand, the Xilinx SRAM radiation hardened FPGA range includes the legacy Virtex-4QV FPGA (90nm) device family, the Virtex-5QV FPGA (65nm) XQR5VFX130 device and the RT Kintex UltraScale (20 nm) XQRKU060 device, which is currently the state-of-the-art in terms of performance.

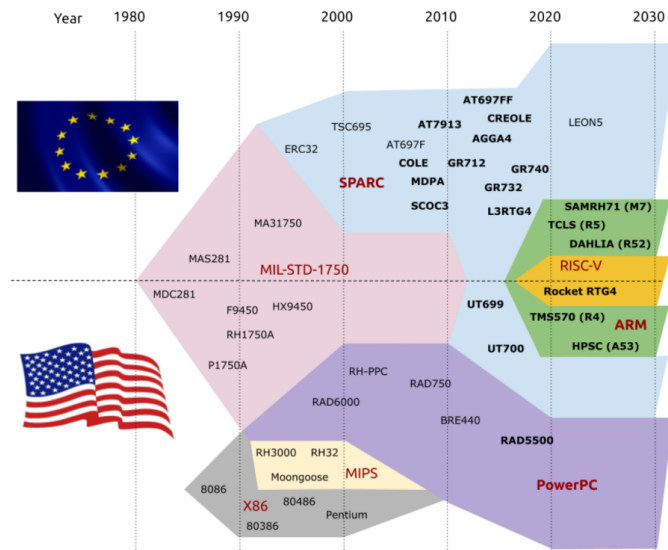
In both cases which offers a wide selection of RHBD FPGAs, which have a long heritage in space missions. With their vendors being based in the USA, however, all these products are subject to USA export controls, which adds insecurity to the European missions' planning. Recently, the NanoXplore family of RHBD devices has been introduced as a European solution, although it has not yet presence in space.

Interestingly, an emerging trend for extending the application area of commercial Xilinx ZynQ and ZynQ Ultrascale+ SoCs into aerospace applications has risen recently. A number of research activities has been focusing on the study of the susceptibility of the ZynQ-7000 series SoCs. The SEU behaviour of the ZynQ-7020 SoC's integrated ARM Processing System is the subject of the work in [26]. In [27],

the authors present their results on heavy proton SEU testing of the same device, while [28] evaluates the SEE behaviour of the NINANO board used in EYE-SAT nanosatellite. The work in [29] is the most complete analysis of the SEE behaviour of ZynQ-7000 series programmable logic and configuration memory under heavy ion irradiation. One of its major contributions is that it provides the tools for the design of efficient mitigation techniques, including effective ECC and configuration memory scrubbing. At the same time, a multitude of research activities incorporate these SoCs for aerospace applications. In [21], for example, we have introduced a high performance parallel implementation of an accelerator for the CCSDS 123.0-B-1 hyperspectral compression algorithm. This work leverages the resources of both the processing system and the programmable logic to deliver state-of-the-art throughput performance. The authors in [30] propose a hybrid convolutional neural network accelerator for semantic segmentation of image, which is widely used in space applications. Their work is evaluated on Xilinx ZynQ and ZynQ Ultrascale+ MPSoCs, while performing error injection and radiation-beam testing, in order to characterise the response of the proposed architectural framework in the presence of radiation phenomena. In all these cases, mostly soft techniques are used as mitigation measures. TMR effectiveness under heavy ion radiation is evaluated in [31] for a ZynQ 7000 SoC supporting a CCSDS 121.090-B-2 compression IP core, demonstrating a 40% increased Mean Time To Failure (MTTF). A rather complete study of the effectiveness of soft methods is presented in [32]. The key takeaway is that for non mission critical systems, soft SEE mitigation techniques can provide the resilience required for space applications.

In DSCAL, the following boards are available and used in the scope of the current thesis:

- The KCU105 evaluation board, built around the Kintex UltraScale XCKU040 device, which is the commercial equivalent of the radiation tolerant Kintex Ultrascale XQRKU060. Regarding the rest of the board's equipment, of notable interest are the two SFP+ cages, which were used for spacefibre integration and the 2 GB of DDR4 RAM memory.
- The ZC706 board, featuring a Zynq-7000 XC7Z045 SoC at speed grade 2, with two ARM Cortex-A9 MPCore hard processors. One SFP+ cage is also included. The board also includes 1 GB of DDR3 RAM connected to the processing system (PS) build around the two ARM processors (component memory), as well as 1 GB of DDR3 RAM for the programmable logic (SODIMM memory). Access to the two memories is independent (both memories can be accessed at the same time).
- The Zedboard, with the Zynq-7000 SoC XC7Z020. The board has no SFP+ connections, but it includes 512MB DDR3 memory connected to the processing system. Access to the memory space from the programmable logic can be



provided from the ZynQ's PS AXI3 high performance (HP) ports, as detailed in Section

2.2 Space-grade CPUs

Similarly to what is described in Section 2.1, the requirements of CPUs are radically different between terrestrial and aerospace applications. Commercial CPUs, target hugely larger market shares, can meet lower time to market requirements and include advanced features and vastly higher performance. Not being able to withstand the harsh environmental conditions typically met in spaceflight, however, they fail to meet the reliability requirements of space missions. Space-qualified CPUs have therefore been developed to mitigate these issues. These CPUs are based on commercial Instruction Set Architectures (ISAs), so the cost of the ecosystem around them is reduced. The ecosystem includes hardware design processes and tools, as well as software tools for applications development. Historically, MIL-STD-1750 architecture [33] dominated space missions, due to its already widespread adoption by military airborne computers. Quickly, however, following the evolution of commercial architectures, the market was dominated by SPARC and PowerPC. A pictorial overview of the historical evolution of space CPUs is provided in Fig.1. The European Space Agency (ESA) opted for the SPARC architecture, mainly because of the widespread availability of software and its open architecture, which allowed the Agency's independence from specific vendors. To this aim, ESA funded the development of the LEON processor in late 1997. One of the principal objectives of the project was the integration of fault tolerant-by-design techniques. The processor should be able to detect and tolerate one error in any register without software

intervention, and to suppress effects from Single Event Transient (SET) errors in combinational logic [35]. LEON evolved in the following years and currently, LEON3 [36] is the most widely adopted platform for ESA missions. LEON3 is distributed as synthesizable VHDL model of a 32-bit processor compliant with the IEEE-1754 (SPARC V8) architecture by Aeroflex Gaisler. The distribution is under the GNU GPL license allowing use for any purpose without licensing fee. The most significant upgrades over the previous LEON2 is the support of Symmetric Multi Processing (SMP) and pipelined operation at 5 stages. In space missions, a fault-tolerant version of the processor (LEON3FT) is the one that is widely used. Fault tolerance is assured by the implementation of ECC coding of all on-chip RAM blocks, which is able to detect and correct up to four errors per 32-bit RAM words or per cache memory tag, and all these without performance impact (completely transparent to user applications). The main means for achieving fault-tolerance is by using ECC coding of all on-chip RAM blocks. A famous LEON3-based SoC is the GR712RC from Aeroflex Gaisler.

LEON5 is the latest version of the LEON processor family[37] and it primarily targets high-end FPGA's. Although it has not yet been implemented in space missions, it provides backward compatibility for most of the software implementations that have targeted LEON3 and LEON4 processors, claiming up to 85% higher performance. Nevertheless, the Reduced Instruction Set Computer version V (RISC-V) Instruction Set Architecture (ISA) is expected to dominate upcoming on-board processing applications [34]. In European missions for example, the DeRISC project [38] has recently shown the first milestones for a multi-core RISC-V processor for aerospace designs. The project is based on the NOEL-V 64-bit RISC-V processor core [39] from Aeroflex Gaisler and state-of-the-art hypervisor technology to accomplish high performance workloads, on a complete processing platform for space. The three processor models (LEON3/5, NOEL-V) are distributed as parts of the open-source GRLIB IP library, which is an integrated set of reusable IP cores, designed for system-on-chip (SoC) development and they are available also in fault-tolerant versions for FPGA and ASIC implementations. Typically, they are interconnected through Advanced Microcontroller Bus Architecture (AMBA) Advanced High-performance Bus (AHB) and Advanced Peripheral Bus (APB) interfaces. A typical SoC built around a single LEON/NOEL processor core with the peripherals included in the distribution is depicted in Fig. 2

As its name implies, the AHB JTAG component shown in the image provides a JTAG debug link to the SoC, allowing among other things the uploading and debugging of user software to the processor's memory, through the GRMON software tool, which is part of the processor's software ecosystem. Other useful software tools included in the ecosystem are the cross-compiler for the CPU architecture and a simulator (TSIM). The debugging capabilities are completed with the Debug Support Unit (DSU) depicted in the image. This module communicates with the CPU through a dedicated debug interface (in addition to AHB) and has complete

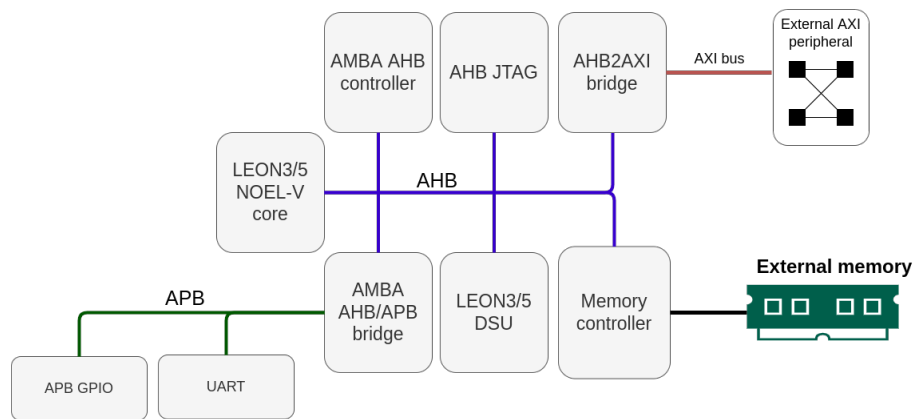


Figure 2: A typical LEON/NOEL SoC design

The next generation of SpaceWire is the SpaceFibre technology, standardised as ECSS-E-ST-50-11C [43]. Except for higher data rates (6,25 Gbps signalling rated), SpaceFibre comes with other significant enhancements:

- Fibre-optic cabling, with electrical support for backwards compatibility with SpaceWire.
- Multi-laning, which can combine the throughput of multiple physical links (lanes) to support well over 20 Gbit/s.
- Advanced Quality of Service (QoS) mechanisms, like prioritization of Virtual Channels, bandwidth reservation and support of deterministic delivery constraints.

DSCAL is a partner of the Hi-SIDE project (<https://www.hi-side.space/>) consortium and as such, it has been granted a access to SpaceFibre test equipment, featuring a STAR-Ultra PCIe interface and link analyzer card (<https://www.star-dundee.com/products/star-ultra-pcie>), along with the necessary software tools (GUI and API for the development of custom applications and performance measurements). An accompanying encrypted IP core netlist, adds transparency in the communication of the software channel interface on the host PC's application with the logic implemented on an FPGA. The core exposes up to 7 AXI4-Stream 128-bit interfaces to the user logic and a RMAP port for configuration and control.

The provided equipment can provide two 4-lane SpaceFibre links to a suitable corresponding interface of up to 20 Gbit/s per link, depending on the number of SFP interfaces of the available FPGA platforms. The maximum data rate can be achieved when all four 6,25 Gbit/s lanes are used. Because of the 8b/10b encoding on the SpaceFibre link, only 80% of the lane bandwidth is available to the user logic. Of the equipment available at DSCAL, only the Zynq UltraScale+ MPSoC ZCU102 development board with 4 SFP+ connector cases can support the maximum data rate of 20 Gbit/s. The ZCU105 card, however, which has been extensively used in this work can only provide up to 10 Gbit/s of user bandwidth.

As part of the Hi-Side project's deliverables, a sample design for the KCU105 had been provided to DSCAL. The reference design included a basic demo for a loopback test through the card's FMC connectors, without connectivity to the host PC. The reference design had to be modified, so that SERDES is mapped to the transceivers allocated to the SFP+ connectors of the board. The transceivers needed also to be parametrized and a suitable clock source to be configured on the board and connected to the transceivers' CPLDs. This process was different for the equipment used in DSCAL: the KCU105 and ZCU102 boards use GTH transceivers, while ZC706 uses GTX transceivers. Figure 4 is a block diagram of the resulting environment used for testing. The design depicted refers to the configuration used on the KCU105 board, but except for the number of lanes and type of transceivers, it is the same for all the other boards.

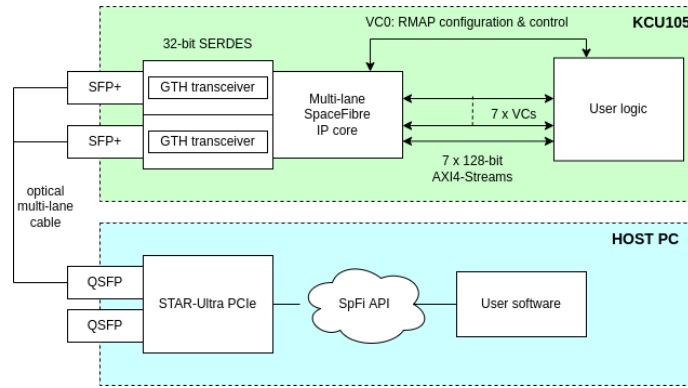


Figure 4: Block diagram of the SpaceFibre equipment and environment

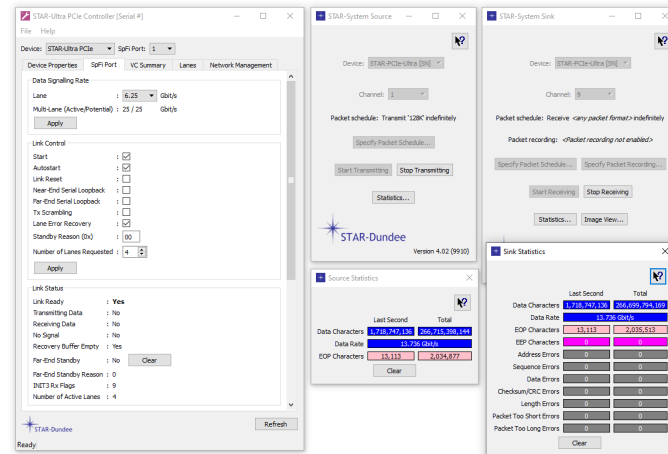


Figure 5: GUI applications examples

On the PC side, the STAR-System software provided as part of the Hi-SIDE equipment provides the drivers of the SpaceFibre STAR-Ultra PCIe board, as well as the software tools for transmitting and receiving packets to and from the SpaceFibre IP code. The software bundle provides two options: a set of GUI applications and a complete API for the development of custom software tools. In both cases, statistics and performance data can be derived. Figure 5 gives an example of the GUI applications. In order to streamline the automatic execution of scripts including SpaceFibre transactions, the following custom applications were developed, based on the STAR-System API:

- `rmapRead < address >`: reads a 4-byte register at a user-specified address.
- `rmapWrite < address > < value >`: writes a 32-bit value at a register at a user-specified address.
- `sendFile -c < channel > -i < file >`: sends a binary file to the specified channel of the master stream interface of the core.
- `sendFile -c < channel > -i < file >`: receives data from the specified channel until an End Of Packet (EOP) character is sent and writes the stream into a file.

2.4 Bit-level channel coding

partial reconfiguration on xupv5 testing with matlab model

2.5 Magnetic recording media coding

2.6 Packet-level coding

Do not forget to add the memory subsystem we created for LEON on ZynQ system (access to DDR memory through

3 Review of the state-of-the-art

3.1 QC LDPC bit-level codes

In general, LDPC encoding refers to the process of calculating the mapping $s \rightarrow c$ of a k -bit binary vector $s \in \{\mathbb{F}_2\}^k$ to the proper element c of the k -dimensional subspace $V \subset \{\mathbb{F}_2\}^n$, according to the code definition, which is defined by the parity-check matrix H of the code, so that the parity-check equation $cH^T = 0$ is satisfied.

The encoding methods for LDPC codes can be classified into the following categories:

Direct method

The direct method involves the application of Gaussian elimination to calculate the generator matrix G from the null space of the parity-check matrix H of the code, that is to solve the equation $GH^T = 0$. This process takes place offline and depending on the encoder's implementation details, the generator matrix data or structure is stored into the encoder. A codeword c can thus be calculated from the input information block s through the vector-matrix multiplication $c = sG$.

In order to facilitate encoder design, for all the practical LDPC codes used in modern communications systems, the generator matrix can be calculated in systematic form. For a (n, k) linear block code in this case $G = [I_k \ W_{n-k}]$, where I_k is the $k \times k$ identity matrix and for QC codes, W_{n-k} is an array of dense cyclic sub-matrices, with the structure of (1). The resulting codeword c is consequently $c = [s \ p]$, where p is the vector of the $n - k$ parity bits. In this case, the encoders implementing this method need only to store the $r \times c \times m$ bits of the first rows (or columns) of these circulants. However, despite the fact that the initial parity-check matrix H is sparse, the resulting $W_{n,k}$ matrix and consequently its constituent circulants are dense matrices. In [44], [45] and [46] compression methods are proposed, which reduce the memory requirements to store the circulants of the parity-check matrix H in the encoder. These methods however are not applicable to dense matrices, nor are the corresponding architectures which handle the sparse matrix operations involved in the calculations.

$$W_{n-k} = \begin{bmatrix} W_{1,1} & \dots & W_{1,c} \\ \vdots & \ddots & \vdots \\ W_{r,1} & \dots & W_{r,c} \end{bmatrix} \quad (1)$$

Encoders proposed in [47],[48],[49],[50],[51],[52],[53] are based on the direct encoding method. My preliminary work in [47] has introduced an efficient architecture for the parallel execution of the vector-matrix multiplication involved in the direct encoding method, by leveraging the inherent parallelism of the generator matrix of CCSDS codes, achieving.

Although the work in [48] focuses in CCSDS codes, the proposed architecture handles encoding inefficiently, requiring large XOR operations over a significant

number of bits ($k/2$ or 2048 in the provided example) and at the same time register resources are wasted. Algorithmically, the approach is also equivalent to [52] and the parallel SRAA of [50]. The required logic resources for hardware implementation are inevitably a large portion of a Virtex7-xc7vx485t FPGA.

In [49], the problematic dimension of the C2 code at the boundaries of the 511-bit circulants when processing a stream of input data is handled by packing input bits present on the 16-bit input bus into groups of 21-bits and unpacking them subsequently to groups of 7 bits for the AND-XOR operations. The difference however between the size of the input bus (16 bits) and the degree of parallelism in the AND-XOR process (MAC module), leads to suboptimal use of the resources of the MAC module, which remains idle for a number of cycles, when input has starved. Moreover, additional computation cycles are wasted by the 18 trailing zeros which are prepended to the information block, according to C2 code description, which are processed by the MAC module in this particular approach. Our architectures handle these problems in a more efficient way, as described in a subsequent section.

The authors in [50] propose various types of encoding circuits, based on shift registers, which achieve encoding complexity linearly proportional to the number of parity bits of the code ($n - k$ according to the notation in this work), or the n total bits of the code in the case of the parallel approach. The SRAA serial encoding scheme described is practically the naive approach provided in the CCSDS standard [6], based on a shift register for the circulants and a register for the calculation of parity bits. The calculated complexity does not include the memory and the necessary circuitry for the loading of the generators $g_{i,j}$ of the circulants to the SRAA shift registers, which incur significant resources cost in practical implementations, as it will be shown in a subsequent section. According to the parallel SRAA approach, which achieves encoding in cb cycles (following the writers' notation), all k input bits participate in the calculation of each parity bit in one clock cycle. This architecture could not be implemented with reasonable resources in practical encoders for codes with block lengths in the range of several thousands of bits and should be considered only as a theoretical approach for academic research purposes only. Even in this case however, the AND-XOR binary calculations on a large number of bits would necessitate large combinatorial paths and would severely jeopardize throughput performance. The two-stage encoding scheme described is practically the *H2-inverse* method described later in the current Section.

The work in [51] proposes an architecture based on Linear Feedback Shift Registers (LFSRs). The input information bits are multiplied with the first rows of the circulants $W_{i,j}$ and instead of rotating the circulant registers, the rotation concerns the output register, which contains the parity bits at the end of the encoding process.

The approach described in [52] is algorithmically equivalent to the parallel SRAA approach of [50], without taking advantage of the QC structure of the targeted codes (IEEE 802.16e). Its performance however is also dominated by the large XOR binary operation involved. In addition, the memory requirements for the storage

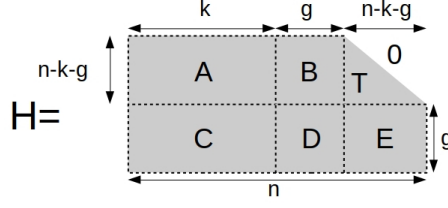


Figure 6: Structure of H matrix for the R-U method

of the generator matrix, totalling $(n-k)k$ bits, pose considerable constraints to the associated hardware. Finally, [53] is another adoption of the SRAA architecture of [50] employing the direct method, optimized for sparse circulants.

R-U method

This method is based on the fact that the codeword can be calculated directly from the H matrix by solving the system of equations defined by the parity-check equation $Hc^T = 0$. The Richardson-Urbanke (R-U) method [54] solves this equation with complexity almost linear to the block length, provided that the parity-check matrix of the corresponding QC-LDPC code has approximate upper-triangular structure, or it can be transformed to such form, which is depicted in Fig.6. For a systematic code with a H matrix of size $(n-k) \times k$, the calculated codeword has the form $c = [s \ p_1 \ p_2]$, where s is the input vector and p_1, p_2 are parity bits vectors of length g and $m-g$ respectively and the parity bits are calculated according to (2), (3), (4).

$$\varphi = ET^{-1}B + D \quad (2)$$

$$p_1^T = \varphi^{-1}(ET^{-1}A + C)s^T \quad (3)$$

$$p_2^T = T^{-1}(As^T + Bp_1^T) \quad (4)$$

The above equations involve many sparse matrices, but only a single dense, namely φ^{-1} . Sparse matrix operations can be implemented by simplified hardware and the determinant factor affecting the performance of the encoder becomes the $g \times g$ dense matrix φ^{-1} . The parity-check matrix of many widely adopted LDPC codes has been specifically designed so that the parameter g is small, or in the case of DVB-S2 is zero and the matrix φ^{-1} has a special structure which results in efficient hardware implementation. For example, the φ matrix of the LDPC codes adopted for IEEE 802.11ac/n, 802.16e and many other applications is the $g \times g$ identity matrix.

For many other codes, transformation into approximate lower triangular form, without affecting the QC structure of the matrix is not straightforward. For example, in the case of the CCSDS codes defined in [6], this can be achieved by shifting the last 4 circulants ($4m$ bits) by 8 columns ($8m$ bits) to the left. Since the last $4m$ bits

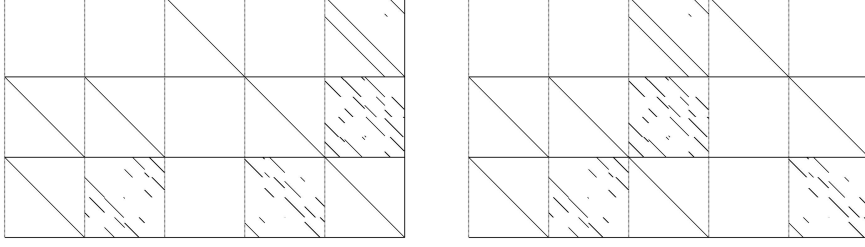


Figure 7: H matrix before (left) and after (right) the transformation into lower triangular form.

of the code are punctured, this permutation does not affect the encoder's output. Fig.7 displays the H matrix before and after the transformation for rate 1/2 AR4JA code with $k = 1024$. The parameter g is therefore $4m$ and φ^{-1} is a $4m \times 4m$ dense QC matrix of $m \times m$ circulants. Architectures proposed in [46], [55], [56], [57], [58], [59] are examples of application of the R-U method.

The work in [46] does not target QC codes, nor can it efficiently handle large dense φ matrices. The parameter g is 2 in the provided implementation examples and the resulting encoders occupy a large amount of the resources of Xilinx XC2V4000-6 FPGA, including a number of Block RAMS. The encoder architecture in [55] targets IEEE 802.11n codes where φ matrix is the identity matrix, but is not applicable to CCSDS codes. Targeting Wimax LDPC codes, [58] also assumes that φ is the identity matrix. In [56] the authors propose a code construction method, together with encoder-decoder architectures. The proposed encoder implements the R-U method, but the code construction aims at minimizing the parameter g . The dense matrix multiplication (3) involving φ in their case is executed on all elements of φ^{-1} in parallel, which obviously does not scale efficiently for large g . The method proposed in [59] and [57] employs SRAA modules introduced in [50] for the dense matrix operations of the RU algorithm. The scalability issues concerning the adoption of SRAA architectures for the direct encoding method, also pertain to the R-U method for CCSDS codes, because of the size of parameter g .

Partitioned H methods

This class of methods is based on the fact that for all systematic codes, the codeword c consists of the systematic part, which is a copy of the input information block and the parity bits: $c = [s \ p]$. The parity-check matrix can therefore be partitioned into a $(n - k) \times k$ submatrix H_1 and a $(n - k) \times (n - k)$ submatrix H_2 , where $H = [H_1 \ H_2]$, so that the parity bits vector can be calculated by (5), (6), (6).

$$Hc^T = H_1s^T + H_2p^T = 0 \quad (5)$$

$$H_2p^T = H_1s^T \quad (6)$$

$$p^T = H_2^{-1} H_1 s^T \quad (7)$$

Submatrix H_1 is sparse and the vector $H_1 s^T$ can be easily calculated. For many practical codes, submatrix H_2^{-1} exhibits regular structure, which facilitates the involved calculations. A common structure in the parity-check matrix of many codes is the dual-diagonal: the rightmost part or of H_2 , or even the entire submatrix (IEEE 802.11 n/ac, 3GPP2 DVB-S2) is a dual-diagonal matrix.

According to a variation of this method ([60], [61]), H_2 matrix is decomposed into a permutation matrix Π and two triangular matrices L, U , using triangular factorization (or LU decomposition). Equation (6) is therefore transformed into (9), from which parity bits are calculated using back-forward substitution. Conversely, the LU decomposition can be applied on H_2^{-1} matrix, so that parity bits are calculated according to (10),(11).

$$H_2 = \Pi^{-1}(LU) \quad (8)$$

$$L[U(p^T)] = \Pi(H_1 s^T) \quad (9)$$

$$H_2^{-1} = \Pi'^{-1}(L'U') \quad (10)$$

$$\Pi' p^T = L'[U'(H_1 s^T)] \quad (11)$$

The architectures proposed in [62], [63], [64], [65], [66] and [67] all follow algorithmically equivalent approaches which assume a dual-diagonal H_2 matrix. Parity bits can be calculated directly from the vector $H_1 s^T$ using backward substitution. In (7), H_2^{-1} is a lower triangular matrix and $H_2^{-1}(i, j) = 1, i \geq j$, so that back substitution is applicable.

For another class of codes (for example in IEEE 802.16e), H_2^{-1} has the approximate dual-diagonal structure of (12), where $I_i^{(x_j)}$ are permutation matrices. Targeting these cases, [68], [69] and [70] propose encoders with similar algorithmical description, which perform necessary permutations along with backward substitution for the calculation of the corresponding parity bits.

$$H_2^{-1} = \left[\begin{array}{c|cccc} I_1^{(x_1)} & I & I & \dots & 0 & 0 \\ \vdots & & & \dots & & \\ I_b^{(x_b)} & 0 & 0 & \dots & I & I \end{array} \right] \quad (12)$$

For many codes however (including those in [6]), H_2^{-1} matrix has the structure of (13), where 0_{4m} and I_{4m} are the $4m \times 4m$ zero and identity matrices and $W_{i,j}$ are $m \times m$ dense circulants, which is obviously not dual-diagonal, rendering the above

architectures altogether inapplicable.

$$H_2^{-1} = \begin{bmatrix} I_{4m} & W_{1,1} & \dots & W_{1,8} \\ 0_{4m} & \vdots & \ddots & \vdots \\ 0_{4m} & W_{12,1} & \dots & W_{12,8} \end{bmatrix} \quad (13)$$

The variation of the method based on L-U decomposition of H_2 according to (8)-(11) is used in [60], where the authors also propose an offline preprocessor for the triangulation of H_2 . The QC structure of the matrix however is not kept in the decomposed matrices, at least for the demonstrated codes. Reference [71] proposes the same encoder architecture with a different decomposition algorithm for CMMB codes, which are not QC. The same encoder architecture is also proposed for CMMB in [45], without details on the decomposition algorithm. The work in [61] targets random Gallager codes. The encoding process is identical to [60], however algorithms are provided for the calculation of permutation matrices, which minimize the density of L , U components. It is shown that the compression achieved for the storage of sparse L , U matrices favors this method over R-U, at least for Gallager codes. The adoption of this encoding method however for CCSDS inflicts a major performance penalty because of the loss of QC structure in L , U matrices. For example, using the triangulation process outlined in [60], the AR4JA rate 1/2 code with $k=1024$ bits calls for the storage, proper indexing and processing of a total of around 64K nonzero elements of L , U matrices, compared to the simple storage of the 2K elements of the first rows of the circulants needed for the $4m \times 4m$ φ^{-1} matrix.

The work in [44] modifies the procedure in [60] and adopts LU decomposition of H_2^{-1} , so that parity bits are calculated according to (11). It is shown that for the selected codes (Multi-level QC-LDPC codes), this method can result in more efficient storage of L' , U' and Π' in the encoder's memory than H_2^{-1} or storing the components of H_2 according to [70]. However, the efficiency of the proposed encoding scheme is limited only to the two-step expanded codes, which results in cyclic structure in the triangulated components. Fig.8 depicts an example of the generated L' and U' matrices, compared to their CCSDS equivalents. The VMM architectures proposed for the vector-matrix multiplications are evidently not applicable for the random matrices of CCSDS. Memory requirements for indexing the non-zero values are also a considerable drawback. Another fully parallel method is also proposed, targeting high throughput. According to this method, there is no need to store L' , U' in memory and the vector-matrix multiplications of (11) are executed in parallel on all bits. This method cannot scale for higher block lengths or other codes. Even for CCSDS AR4JA $k=1024$, rate 1/2 code, the implementation of this method required more than 72K LUTs, fitting only high-end Virtex 5 FPGAs, with prohibitive routing delays for any practical application.

In [72] the authors propose a hybrid approach, according to which the parity-check matrix is transformed in approximate lower-triangular form, as in R-U method. The parity bits are calculated using a mix of the direct and the R-U method: The

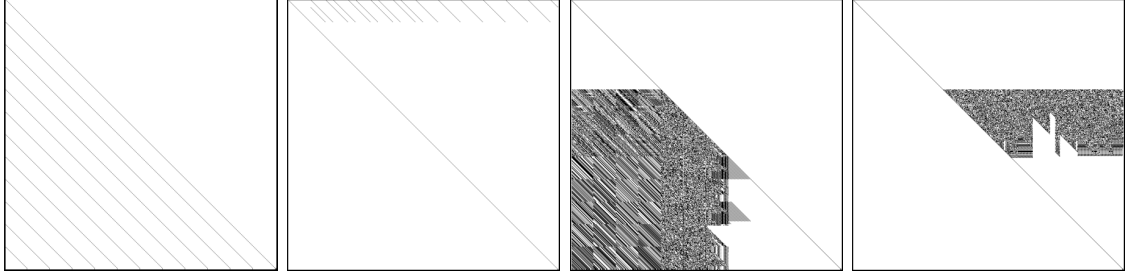


Figure 8: L' , U' matrices of (2016,1008) example code in [44] (left) and AR4JA $k=1024$ rate 1/2 code (right).

first subvector p_1 of g parity bits in R-U equation (3) is calculated from the the generator matrix G , according to the direct method. In this case, only the first g columns of the submatrix W_{n-k} in (1) need to be stored in encoder memory, avoiding thus the dense vector-matrix operations involving φ^{-1} . The paper focuses in IEEE 802.11an codes. For CCSDS AR4JA codes however, $g = 4m$ and parameter r is 8, 16, 32 for rates 1/2, 2/3, 3/4 correspondingly, while φ^{-1} is always $4m \times 4m$. No performance gain is therefore achievable from this method for CCSDS codes. On the contrary, memory requirements and critical path are adversely affected from the larger dense matrix involved. Also T^{-1} is the identity matrix for AR4JA and the critical path is not affected.

3.2 Packet level erasure codes

Packet-level erasure coding has been proposed for many modern applications, such as edge computing [73], underwater acoustic sensor networks [74], magnetic recording media [75], hybrid broadcasting broadband television (HbbTV) [76] and delay tolerant networks (DTN) over deep space communication systems [77]. Joint use of erasure coding and bit-level FEC schemes in different scenarios has been studied in [78, 79, 80].

The main focus of this thesis is on the implementation of encoders for the codes defined in [12]. The only implementation of the proposed codes is their integration into the the Interplanetary Overlay Network (ION) software suite [81], which is a software implementation of the bundle protocol for Delay Tolerant Networks (DTN). In [77], a multithreaded implementation of the ION libraries is proposed for better performance. However, the purely software approach proposed is expected to exert considerable strain on the on-board general purpose processor and mass memory subsystem of a space Software Defined Radio (SDR), which is typically responsible for these functions. Offloading these tasks to a small footprint hardware accelerator integrated into a FPGA is especially important in the case of microsats and cubesats and high data rate optical communications, to achieve reduced size, weight, power, and cost (SWaP-C). Typically, spacecraft subsystems already include

FPGAs responsible for command and data handling (C&DH) tasks and the recent trend is to fully utilize these devices for multiple combined functions [82]. Moreover, FPGA hardware acceleration of packet-level coding enables a very high speed data processing chain providing data rates in the scale of several Gbps.

4 Goal of Thesis

In the proposed thesis, I provide my research results on hardware implementations of encoders for the two cases described above: QC LDPC codes with no specific structure (other than QC) in the parity check matrix and the packet level erasure codes defined in [12]. Regarding bit-level LDPC codes, my work introduces a novel solution, optimized for the codes of [6], the most important feature of which is the efficient bit vector multiplication with dense matrices. Such multiplications are key operations for all LDPC encoding methods. This solution enables the design of novel encoder architectures and the resulting hardware implementations can achieve throughput performance in the range of multiple Gbps, with low resource utilisation. A lot of discussion is taking place lately about the endorsement of the rate 1/2 LDPC codes of [6] into the new optical space communication standards [7]. The required performance however of the LDPC encoding and decoding components of the codec remains a challenging task and an active research area.

At the same time, the current work is the first approach to examine packet-level encoding algorithms and propose, implement and test hardware encoder architectures for these algorithms. Since their introduction in [12], the proposed codes have not matured into a CCSDS recommended ("blue") standard yet. With the current research, I support that they can be placed among the options for modern high speed communications.

The theoretical results and analytical estimations are in all cases backed by active development and implementations on FPGA and MPSoC hardware, which also include validation and verification procedures: the proposed architectures are implemented as IP cores on the targeted platforms and their responses are compared against a bit-accurate software model, written in C or GNU/Octave. This development and testing process accounts for a significant part of the total research effort and calls for solid understanding and proficient use of the corresponding tools:

- Xilinx Vivado
- Mentor Graphins Modelsim simulator
- Synopsis Synplify
- Vunit framework [83]
- GNU/Octave language, which is the open-source equivalent of Mathworks Matlab

The DSCAL equipment available to support the research includes all the above listed software and a variety of development boards, including the XUPv5, Zedboard, ZC706 and ZCU102 boards.

References

- [1] W. Ryan and S. Lin, *Channel Codes: Classical and Modern*. Cambridge University Press, 2009.
- [2] I. S. Reed and G. Solomon, “Polynomial Codes Over Certain Finite Fields,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, jun 1960.
- [3] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding: Turbo-codes. 1,” in *Proceedings of ICC '93 - IEEE International Conference on Communications*, vol. 2, 1993, pp. 1064–1070 vol.2.
- [4] R. Gallager, “Low-density parity-check codes,” *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, jan 1962. [Online]. Available: <http://ieeexplore.ieee.org/document/1057683/>
- [5] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.
- [6] *TM Synchronization and Channel Coding*, CCSDS Recommended Standard 131.0-B-1, Sep. 2017. [Online]. Available: <https://public.ccsds.org/Pubs/131x0b3e1.pdf>
- [7] CCSDS141.11-O-1, “Optical High Data Rate (Hdr) Communication 1064 nm Optical High Data Rate (Hdr) Communication ,” CCSDS, Tech. Rep. December, 2018.
- [8] CCSDS142.0-B-1, “Optical Communications Coding and Communications Coding and,” 2018.
- [9] N. Tang and Y. Lin, “Fast Encoding and Decoding Algorithms for Arbitrary F2m,” *IEEE Communications Letters*, vol. 24, no. 4, pp. 716–719, apr 2020.
- [10] R. Adhikary, J. N. Daigle, and L. Cao, “Dynamic Code Selection Method for Content Transfer in Deep-Space Network,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, pp. 456–474, feb 2020.
- [11] A. Guillén i Fàbregas, “Coding in the block-erasure channel,” *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 5116–5121, nov 2006.
- [12] *Erasur Correcting Codes for Use in Near-Earth and Deep Space Communications*, CCSDS Experimental Specification 131.5-O-1, Nov. 2014. [Online]. Available: <https://public.ccsds.org/Pubs/142x0b1.pdf>
- [13] *ECSS-E-HB-11A Space engineering Technology readiness level (TRL) guidelines*, ECSS guidelines, Mar. 2017. [Online]. Available: https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=37110
- [14] R. Ecoffet, “Overview of in-orbit radiation induced spacecraft anomalies,” *IEEE Transactions on Nuclear Science*, vol. 60, no. 3, pp. 1791–1815, 2013.
- [15] J. S. George, “An overview of radiation effects in electronics,” *AIP Conference Proceedings*, vol. 2160, no. 1, p. 060002, oct 2019. [Online]. Available: <https://aip.scitation.org/doi/abs/10.1063/1.5127719>
- [16] Q. Huang and J. Jiang, “An overview of radiation effects on electronic devices under severe accident conditions in npps, rad-hardened design techniques and simulation tools,” *Progress in Nuclear Energy*, vol. 114, pp. 105–120, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0149197019300563>
- [17] G. Furano and A. Menicucci, *Roadmap for On-Board Processing and Data Handling Systems in Space*. Cham: Springer International Publishing, 2018, pp. 253–281.

- [Online]. Available: https://doi.org/10.1007/978-3-319-54422-9_10
- [18] P.-P. Mathieu, M. Borgeaud, Y.-L. Desnos, M. Rast, C. Brockmann, L. See, R. Kapur, M. Mahecha, U. Benz, and S. Fritz, “The esa’s earth observation open science program [space agencies],” *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 2, pp. 86–96, 2017.
 - [19] D. L. Oltrogge and K. Leveque, “An Evaluation of CubeSat Orbital Decay,” in *25 th Annual AIAA/USU Conference on Small Satellites*, 2011.
 - [20] G. Lentaris, K. Maragos, I. Stratakis, L. Papadopoulos, O. Papanikolaou, D. Soudris, M. Lourakis, X. Zabulis, D. Gonzalez-Arjona, and G. Furano, “High-performance embedded computing in space: Evaluation of platforms for vision-based navigation,” *Journal of Aerospace Information Systems*, vol. 15, no. 4, pp. 178–192, 2018.
 - [21] A. Tsigkanos, N. Kranitis, D. Theodoropoulos, and A. Paschalis, “High-Performance COTS FPGA SoC for Parallel Hyperspectral Image Compression with CCSDS-123.0-B-1,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 11, pp. 2397–2409, nov 2020.
 - [22] J. Vidmar, P. Maillard, T. Jones, M. Sawant, G. Gambardella, and N. Fraser, “Space DPU: Constructing a Radiation-Tolerant, FPGA-based Platform for Deep Learning Acceleration on Space Payloads,” in *2nd European Workshop on On-Board Data Processing (OBDP 2021)*, jun 2021. [Online]. Available: <https://zenodo.org/record/5639613>
 - [23] V. Leon, I. Stamoulias, G. Lentaris, D. Soudris, D. Gonzalez-Arjona, R. Domingo, D. M. Codinachs, and I. Conway, “Development and testing on the european space-grade brave fpgas: Evaluation of ng-large using high-performance dsp benchmarks,” *IEEE Access*, vol. 9, pp. 131 877–131 892, 2021.
 - [24] M. J. Marinella, “Radiation effects in advanced and emerging nonvolatile memories,” *IEEE Transactions on Nuclear Science*, vol. 68, no. 5, pp. 546–572, 2021.
 - [25] “PolarFire SONOS Technology Microsemi.” [Online]. Available: <https://www.microsemi.com/blog/2018/04/10/polarfire-sonos-technology/>
 - [26] D. M. Hiemstra and V. Kirischian, “Single event upset characterization of the zynq-7000 arm cortex-a9 processor unit using proton irradiation,” *IEEE Radiation Effects Data Workshop*, vol. 2015-November, 11 2015.
 - [27] L. A. Tambara, F. L. Kastensmidt, N. H. Medina, N. Added, V. A. P. Aguiar, F. Aguirre, E. L. A. Macchione, and M. A. G. Silveira, “Heavy ions induced single event upsets testing of the 28 nm xilinx zynq-7000 all programmable soc,” in *2015 IEEE Radiation Effects Data Workshop (REDW)*, 2015, pp. 1–6.
 - [28] F. Bezerra, D. Dangla, F. Manni, J. Mekki, D. Standarovski, R. G. Alia, M. Brugger, and S. Danzeca, “Evaluation of an alternative low cost approach for see assessment of a soc,” in *2017 17th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*, 2017, pp. 1–5.
 - [29] V. Vlagkoulis, A. Sari, J. Vrachnis, G. Antonopoulos, N. Segkos, M. Psarakis, A. Tavoularis, G. Furano, C. Boatella Polo, C. Poivey, V. Ferlet-Cavrois, M. Kastriotou, P. Fernandez Martinez, R. G. Alia, K.-O. Voss, and C. Schuy, “Single event effects characterization of the programmable logic of xilinx zynq-7000 fpga using very/ultra high-energy heavy ions,” *IEEE Transactions on Nuclear Science*, vol. 68, no. 1, pp. 36–45, 2021.

- [30] S. Sabogal, A. George, and G. Crum, “Recon: A reconfigurable cnn acceleration framework for hybrid semantic segmentation on hybrid socs for space applications,” in *2019 IEEE Space Computing Conference (SCC)*, 2019, pp. 41–52.
- [31] A. Snchez, Y. Barrios, L. Santos, and R. Sarmiento, “Evaluation of tmr effectiveness for soft error mitigation in shyloc compression ip core implemented on zynq soc under heavy ion radiation,” in *2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, 2019, pp. 1–4.
- [32] O. O. Kibar, P. Mohan, P. Rech, and K. Mai, “Evaluating the impact of repetition, redundancy, scrubbing, and partitioning on 28-nm fpga reliability through neutron testing,” *IEEE Transactions on Nuclear Science*, vol. 66, no. 1, pp. 248–254, 2019.
- [33] *Sixteen-Bit Computer Instruction Set Architecture*, DoD Military Standard MIL-STD-1750, Aug. 1980. [Online]. Available: https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=37110
- [34] S. Di Mascio, A. Menicucci, E. Gill, G. Furano, and C. Monteleone, “Leveraging the openness and modularity of risc-v in space,” *Journal of Aerospace Information Systems*, vol. 16, no. 11, pp. 454–472, jan 2020. [Online]. Available: www.aiaa.org/randp.
- [35] J. Andersson, J. Gaisler, and R. Weigand, “Next generation multipurpose microprocessor,” in *Data Systems In Aerospace 2010 (DASIA2010)*, 2010.
- [36] “LEON3.” [Online]. Available: <https://www.gaisler.com/index.php/products/processors/leon3>
- [37] *LEON5 processor*, Cobham Gaisler. [Online]. Available: <https://www.gaisler.com/index.php/products/processors/leon5>
- [38] “De-RISC: Dependable Real-time Infrastructure for Safety-critical Computer.” [Online]. Available: <https://cordis.europa.eu/project/id/869945>
- [39] *NOEL-V Processor*, Cobham Gaisler. [Online]. Available: <https://www.gaisler.com/index.php/products/processors/noel-v>
- [40] A. Athavale and C. Christensen, *High-Speed Serial I/O Made Simple*, 1st ed. Xilinx, 2005. [Online]. Available: www.xilinx.com/xcell
- [41] S. Parkes, *SpaceWire User’s Guide*. Star-Dundee, 2012.
- [42] “SpaceWire Links, nodes, routers and networks,” 2019. [Online]. Available: <https://ecss.nl/standard/ecss-e-st-50-12c-rev-1-spacewire-links-nodes-routers-and-networks-15-may-2019/>
- [43] “SpaceFibre Very high-speed serial link,” 2019. [Online]. Available: <https://ecss.nl/standard/ecss-e-st-50-11c-spacefibre-very-high-speed-serial-link/>
- [44] A. Mahdi and V. Paliouras, “A Low Complexity-High Throughput QC-LDPC Encoder,” *IEEE Transactions on Signal Processing*, vol. 62, no. 10, pp. 2696–2708, may 2014. [Online]. Available: <http://ieeexplore.ieee.org/document/6781047/>
- [45] P. Wang and Y.-e. Chen, “Low-Complexity Real-Time LDPC Encoder Design for CMMB,” in *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, aug 2008, pp. 1209–1212. [Online]. Available: <http://ieeexplore.ieee.org/document/4604260/>
- [46] D.-U. Lee, W. Luk, C. Wang, C. Jones, M. Smith, and J. Villasenor, “A Flexible Hardware Encoder for Low-Density Parity-Check Codes,” in *12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*. , 2004, pp.

- 101–111. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.9061&rep=rep1&type=pdf>
- [47] D. Theodoropoulos, N. Kranitis, and A. Paschalis, “An efficient LDPC encoder architecture for space applications,” in *2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS)*. IEEE, jul 2016, pp. 149–154. [Online]. Available: <http://ieeexplore.ieee.org/document/7604689/>
 - [48] Q. W. Zhaohui Wang, Xin Hao, Changxing Lin, “An Efficient Hardware LDPC Encoder Based on Partial Parallel Structure for CCSDS,” in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*. Chongqing: IEEE, 2018, pp. 136–139. [Online]. Available: <https://ieeexplore.ieee.org/document/8599970>
 - [49] L. Miles, J. Gambles, G. Maki, W. Ryan, and S. Whitaker, “An 860-Mb/s (8158,7136) Low-Density Parity-Check Encoder,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 8, pp. 1686–1691, aug 2006. [Online]. Available: <http://ieeexplore.ieee.org/document/1661745/>
 - [50] Zongwang Li, Lei Chen, Lingqi Zeng, S. Lin, and W. Fong, “Efficient encoding of quasi-cyclic low-density parity-check codes,” *IEEE Transactions on Communications*, vol. 54, no. 1, pp. 71–81, jan 2006. [Online]. Available: <http://ieeexplore.ieee.org/document/1576951/>
 - [51] K. Andrews, S. Dolinar, and J. Thorpe, “Encoders for block-circulant LDPC codes,” in *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005*. IEEE, 2005, pp. 2300–2304. [Online]. Available: <http://ieeexplore.ieee.org/document/1523758/>
 - [52] H. Yasotharan and A. C. Carusone, “A flexible hardware encoder for systematic low-density parity-check codes,” in *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*. IEEE, aug 2009, pp. 54–57. [Online]. Available: <http://ieeexplore.ieee.org/document/5236155/>
 - [53] S.-W. Yen, S.-Y. Hung, C.-L. Chen, H.-C. Chang, S.-J. Jou, and C.-Y. Lee, “A 5.79-Gb/s Energy-Efficient Multirate LDPC Codec Chip for IEEE 802.15.3c Applications,” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 9, pp. 2246–2257, sep 2012. [Online]. Available: <http://ieeexplore.ieee.org/document/6198294/>
 - [54] T. J. Richardson and R. L. Urbanke, “Efficient Encoding of Low-Density Parity-Check Codes,” *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 47, no. 2, 2001. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.123.8553&rep=rep1&type=pdf>
 - [55] G. Tzimpragos, C. Kachris, D. Soudris, and I. Tomkos, “A low-complexity implementation of QC-LDPC encoder in reconfigurable logic,” in *2013 23rd International Conference on Field programmable Logic and Applications*. IEEE, sep 2013, pp. 1–4. [Online]. Available: <http://ieeexplore.ieee.org/document/6645587/>
 - [56] Hao Zhong and Tong Zhang, “Block-LDPC: a practical LDPC coding system design approach,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 4, pp. 766–775, apr 2005. [Online]. Available: <http://ieeexplore.ieee.org/document/1417070/>
 - [57] S. Yu, C. Liu, P. Zhang, and L. Jiang, “Efficient encoding of QC-LDPC codes with multiple-diagonal parity-check structure,” *Electronics Letters*, vol. 50, no. 4, pp.

- 320–321, feb 2014. [Online]. Available: <https://digital-library.theiet.org/content/journals/10.1049/el.2013.2390>
- [58] X. Wang, T. Ge, J. Li, C. Su, and F. Hong, “Efficient Multi-rate Encoder of QC-LDPC Codes Based on FPGA for WIMAX Standard,” *Chinese Journal of Electronics*, vol. 26, no. 2, pp. 250–255, mar 2017. [Online]. Available: <http://digital-library.theiet.org/content/journals/10.1049/cje.2017.01.006>
 - [59] Haibin Zhang, Jia Zhu, Huifeng Shi, and Dawei Wang, “Layered Approx-Regular LDPC: Code Construction and Encoder/Decoder Design,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 2, pp. 572–585, mar 2008. [Online]. Available: <http://ieeexplore.ieee.org/document/4432803/>
 - [60] Jia-ning Su, Hou Iang, K. iu, I. yang eng, Ao Min, and Ao Min, “An Efficient Low Complexity LDPC Encoder Based On Factorization With Pivoting,” in *2005 6th International Conference on ASIC*, vol. 1. IEEE, 2005, pp. 168–171. [Online]. Available: <http://ieeexplore.ieee.org/document/1611277/>
 - [61] Y. Kaji, “Encoding LDPC Codes Using the Triangular Factorization,” *IEICE Trans. Fundamentals Electron., Commun. Comput.Sci*, no. 10, 2006.
 - [62] M. Gomes, G. Falcao, A. Sengo, V. Ferreira, V. Silva, and M. Falcao, “High throughput encoder architecture for DVB-S2 LDPC-IRA codes,” in *2007 Internatonal Conference on Microelectronics*. IEEE, dec 2007, pp. 271–274. [Online]. Available: <http://ieeexplore.ieee.org/document/4497709/>
 - [63] A. A. Al Hariri, F. Monteiro, L. Sieler, and A. Dandache, “A high throughput configurable parallel encoder architecture for Quasi-Cyclic Low-Density Parity-Check Codes,” in *2013 IEEE 19th International On-Line Testing Symposium (IOLTS)*. IEEE, jul 2013, pp. 163–166. [Online]. Available: <http://ieeexplore.ieee.org/document/6604069/>
 - [64] Zhiyong He, S. Roy, and P. Fortier, “Encoder architecture with throughput over 10 Gbit/sec for quasi-cyclic LDPC codes,” in *2006 IEEE International Symposium on Circuits and Systems*. IEEE, 2006, p. 4. [Online]. Available: <http://ieeexplore.ieee.org/document/1693323/>
 - [65] A. A. A. Hariri, F. Monteiro, L. Sieler, and A. Dandache, “Configurable and high-throughput architectures for Quasi-cyclic low-density parity-check codes,” in *2014 21st IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, dec 2014, pp. 790–793. [Online]. Available: <http://ieeexplore.ieee.org/document/7050104/>
 - [66] J. M. Pérez and V. Fernández, “3GPP2/802.20 RC/QC-LDPC encoding,” in *2010 European Wireless Conference, EW 2010*, 2010.
 - [67] Yongmin Jung, Chulho Chung, Jaeseok Kim, and Yunho Jung, “7.7Gbps encoder design for IEEE 802.11n/ac QC-LDPC codes,” in *2012 International SoC Design Conference (ISOCC)*. IEEE, nov 2012, pp. 215–218. [Online]. Available: <http://ieeexplore.ieee.org/document/6407078/>
 - [68] N. A. F. Neto, J. R. S. de Oliveira, W. L. A. de Oliveira, and J. C. N. Bittencourt, “VLSI architecture design and implementation of a LDPC encoder for the IEEE 802.22 WRAN standard,” in *2015 25th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. IEEE, sep 2015, pp. 71–76. [Online]. Available: <http://ieeexplore.ieee.org/document/7347589/>

- [69] S. Kopparthi and D. M. Gruenbacher, "Implementation of a Flexible Encoder for Structured Low-Density Parity-Check Codes," in *2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*. IEEE, aug 2007, pp. 438–441. [Online]. Available: <http://ieeexplore.ieee.org/document/4313268/>
- [70] Chia-Yu Lin, Chih-Chun Wei, and Mong-Kai Ku, "Efficient encoding for dual-diagonal structured LDPC codes based on parity bit prediction and correction," in *APCCAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems*. IEEE, nov 2008, pp. 1648–1651. [Online]. Available: <http://ieeexplore.ieee.org/document/4746353/>
- [71] Xiangran Sun and Dongxin Shi, "Design and optimization of LDPC encoder based on LU decomposition with simulated annealing," in *2011 International Conference on Computer Science and Service System (CSSS)*. IEEE, jun 2011, pp. 2181–2184. [Online]. Available: <http://ieeexplore.ieee.org/document/5974805/>
- [72] A. Cohen and K. Parhi, "A Low-Complexity Hybrid LDPC Code Encoder for IEEE 802.3an (10GBase-T) Ethernet," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 4085–4094, oct 2009. [Online]. Available: <http://ieeexplore.ieee.org/document/4915776/>
- [73] L. Liang, H. He, J. Zhao, C. Liu, Q. Luo, and X. Chu, "An erasure-coded storage system for edge computing," *IEEE Access*, vol. 8, pp. 96 271–96 283, 2020.
- [74] K. S. Geethu and A. V. Babu, "Performance analysis of erasure coding based data transfer in Underwater Acoustic Sensor Networks," in *2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015*. Institute of Electrical and Electronics Engineers Inc., sep 2015, pp. 2145–2151.
- [75] Y. Han and W. E. Ryan, "Packet-LDPC codes for tape drives," *IEEE Transactions on Magnetics*, vol. 41, no. 4, pp. 1340–1347, apr 2005.
- [76] F. Mattoussi, M. Crussiere, J. F. Helard, and G. Zaharia, "Analysis of Coding Strategies Within File Delivery Protocol Framework for HbbTV Based Push-VoD Services over DVB Networks," *IEEE Access*, vol. 7, pp. 15 489–15 508, 2019.
- [77] N. Alessi, C. Caini, T. De Cola, and M. Raminella, "Packet Layer Erasure Coding in Interplanetary Links: The LTP Erasure Coding Link Service Adapter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, pp. 403–414, feb 2020.
- [78] T. A. Courtade and R. D. Wesel, "Optimal allocation of redundancy between packet-level erasure coding and physical-layer channel coding in fading channels," *IEEE Transactions on Communications*, vol. 59, no. 8, pp. 2101–2109, aug 2011.
- [79] P. Ostovari and J. Wu, "Reliable broadcast with joint forward error correction and erasure codes in wireless communication networks," in *Proceedings - 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2015*. Institute of Electrical and Electronics Engineers Inc., dec 2015, pp. 324–332.
- [80] C. R. Berger, S. Zhou, Y. Wen, P. Willett, and K. Pattipati, "Optimizing joint erasure- and error-correction coding for wireless packet transmissions," *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4586–4595, nov 2008.
- [81] "ION-DTN." [Online]. Available: <https://sourceforge.net/projects/ion-dtn/>
- [82] F. Davarian, A. Babuscia, J. Baker, R. Hodges, D. Landau, C.-w. Lau, N. Lay, M. Angert, and V. Kuroda, "Improving Small Satellite Communications in Deep

- SpaceA Review of the Existing Systems and Technologies With Recommendations for Improvement. Part I: Direct to Earth Links and SmallSat Telecommunications Equipment,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 7, pp. 8–25, jul 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/9133660/>
- [83] L. Asplund, “VUnit: a test framework for HDL VUnit documentation,” 2020. [Online]. Available: <https://vunit.github.io/>