DOI: 10.35598/mcfpga.2025.028

Implementation of SDR on FPGA: Methods of Encoding and Decoding in Real Time

Oleksandr Vorgul
Scientific adviser
ORCID 0000-0002-7659-8796
dept.Microprocessor Technologies and System
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
oleksandr.vorgul@nure.ua

Ivan Ihnatiuk
dept.Microprocessor Technologies and System
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
ivan.ihnatiuk@nure.ua

***Abstract*– This report presents a comprehensive analysis of the implementation of key information coding techniques in software-defined radio (SDR) systems based on platforms with field-programmable logic integrated circuits (FPGAs). Three fundamental classes of algorithms are considered in detail: source coding, which provides efficient data compression; channel coding (FEC), which is aimed at increasing noise immunity; and cryptographically strong coding, which guarantees information security. Particular attention is paid to the unique architectural advantages of FPGAs, such as massive parallelism, low latency and energy efficiency , which make them indispensable for real-time signal processing. Practical aspects of implementation in modern communication standards (5G NR, Wi-Fi 6/7, DVB-S2/X), as well as task distribution strategies in hybrid SoC systems are analyzed. The critical role of FPGAs in ensuring the required performance for resource-intensive algorithms, especially in FEC decoders, is proven.**

Keywords— Software-Defined Radio (SDR), FPGA, Channel Coding, Forward Error Correction (FEC), Source Coding, Low-Density Parity Check (LDPC) Codes, Turbo Codes , Hardware Acceleration.

I. Introduction

Software-Defined Radio (SDR) is a paradigm for building radio systems in which traditional hardware components (filters, modulators, demodulators) are replaced by software-implemented functions. This flexibility allows a single hardware platform to support multiple communication standards by simply changing the software. Field-programmable gate arrays (FPGAs) have become the hardware platform of choice for implementing SDR due to their reconfigurability, massively parallel processing capability, low latency, and high energy efficiency when performing specialized computations [3, 10].

Information coding is an integral part of any modern communication system. It covers three interrelated but functionally different processes:

1. Source Coding (Source Coding: Data compression to minimize redundancy and save channel bandwidth.
2. Channel Coding / Forward Error Correction (Channel Coding/ Forward Error Correction, FEC): Introduction of controlled redundancy to detect and correct errors that occur during transmission over a noisy channel.
3. Cryptographically stable coding (Cryptographic Coding: Ensuring confidentiality, authentication and integrity of transmitted data.

Efficient implementation of these methods on limited hardware resources, especially under real-time requirements (low latency) and high throughput (high throughput), is a significant engineering challenge. The purpose of this report is a comprehensive analysis of architectural solutions, algorithmic features and practical applicability of coding methods when implemented in FPGA-based SDR systems. The work systematizes the advantages and limitations of FPGA for each coding class, considers specific examples from modern standards and predicts development trends.

II. Classification and Implementation of Coding Methods on FPGA

1.1 Source Coding: Efficiency through Compression

The main goal of source coding is to reduce the amount of transmitted data without significant loss of information (or with controlled loss), which is critical for systems with limited spectral efficiency, such as satellite communications, mobile networks or tactical radios.

- Typical Algorithms and Standards:
  - Audio: ADPCM (Adaptive Differential PCM), ITU-T G.711 (μ-law / A-law), MP3 (MPEG-1 Audio Layer III), AAC (Advanced Audio Coding), Opus (open, low latency).
  - Speech: Linear Prediction (LPC), CELP (Code-Excited Linear Prediction), ACELP (Algebraic CELP) - are widely used in GSM, UMTS, VoLTE standards .
  - Video: H.264/AVC, H.265/HEVC (High Efficiency Video Coding), VP9 - intensive calculations that often require hardware acceleration.
  - General information: Lossless compression algorithms (LZ77, Huffman coding), often integrated into codecs or higher-level protocols [9].
- FPGA Implementation: Advantages and Strategies:
- Implementing source codecs on FPGA is especially useful in systems with strict latency requirements (voice, video conferencing) or very high throughput/efficiency [3]. Key aspects:

- Massive Parallelism: FPGAs allow efficient parallelization of independent data blocks (e.g. audio frames, video macroblocks), significantly increasing performance.
- Low and Predictable Latency: Direct hardware implementation eliminates the non-deterministic latencies inherent in operating systems and CPU/DSP software stacks, which is critical for interactive applications.
- Power efficiency: Performing computationally intensive compression operations (such as discrete cosine transform - DCT in video codecs) on FPGAs is often more power efficient than on general-purpose processors.
- Hybrid Approach: For complex codecs (H.265), it is typical to separate tasks: resource-intensive calculations (DCT, quantization, vector motion) are performed on the FPGA, and data flow control, buffering and high-level control of the codec are performed on the central processing unit (CPU) as part of a SoC platform (e.g. Xilinx Zynq , Intel Agilex SoC) [6, 7].
- Resource intensity: Implementation of full-fledged video codecs (especially H.265) requires significant FPGA resources (logical cells, BRAM memory blocks, DSP slices), which must be taken into account during design. Ready-made IP cores from FPGA manufacturers or optimized HDL implementations are often used.

1.2. Channel Encoding (FEC): The Foundation of Reliable Communications

FEC is the foundation of robust communications in noisy and disruptive environments. Without it, many modern wireless systems would be impossible or highly unreliable.

- Typical Algorithms and Evolution:
  - Block codes: Hamming codes (single-error correction), BCH (Bose – Chaudhuri – Hocquenghem), Reed-Solomon (RS) — are widely used, often in combination with other methods (e.g. RS + convolutional code in DVB-S).
  - Convolutional codes: Classical method with Viterbi decoding (Viterbi Decoding), which has a good complexity/efficiency ratio.
  - Turbo codes (Turbo Codes): Revolutionized in the late 90s, approaching the Shannon limit. Widely used in 3G (UMTS) and 4G (LTE) systems [8].
  - Low-Density Parity Check (LDPC) codes: The dominant method in modern standards due to its high error-correction capability and suitability for parallel implementation. They are the basis of FEC in 5G NR (data channel), Wi-Fi 6/7 (802.11ax/ be), DVB-S2/S2X/T2 [1, 4, 5].
  - Polar codes (Polar Codes): Used for the control channel in 5G NR, as they theoretically reach the Shannon limit for binary symmetric channels [4].
- FPGA Implementation: Critical Need:
- FEC implementation, especially decoding, is one of the most resource-intensive tasks in the SDR physical layer (PHY). FPGAs are indispensable here for several reasons:
  - Extreme Computational Complexity: Decoding algorithms of modern codes (LDPC, Turbo, Viterbi for long constraints) require a huge number of operations per second at gigabit speeds. Only FPGAs or ASICs are able to provide the required performance [1, 2].
  - Massive Parallelism: This is a key advantage of FPGAs. For example, an LDPC decoder can process hundreds or even thousands of Tanner graph nodes (variable or check nodes) simultaneously in a single clock cycle. A Viterbi decoder can efficiently parallelize the computation of path labels. A turbo code decoder (MAP, SOVA) can parallelize block segments or independent composite decoders [1, 3].
  - Achieving Gigabit Speeds: The requirements of modern standards (e.g. peak speeds in 5G) dictate the need to process data flows in gigabits per second. Hardware implementation on FPGA is the only possible solution for LDPC decoders and turbo codes at such speeds [1, 4].
  - Low and Deterministic Latency: FPGA hardware decoders provide predictably low and consistent processing latency, which is critical for closed-loop systems and real-time applications.
  - Flexibility and Adaptability: FPGAs allow dynamic switching between different FEC schemes ("on the fly") depending on the selected SDR operating mode (e.g. switching between Wi-Fi 6 and LTE) or the current communication channel conditions (when using adaptive coding and modulation - ACM) [3].
  - Resource Intensity and Use of DSP Slices: Implementing efficient FEC decoders consumes a significant portion of FPGA resources. Embedded DSP slices are especially important (Digital Signal Processing slices), optimized for multiply-accumulate (MAC) operations, which are widely used in Viterbi algorithms, MAP decoders of turbo codes, and computations in LDPC nodes [2, 3].

1.3. Crypto-Resistant Coding: A Guarantee of Security

In the face of growing threats to information security, crypto-resistant coding has become a mandatory element of almost all communication systems.

- Typical Algorithms:
  - Symmetric ciphers: AES (Advanced Encryption Standard) - the most common, ChaCha20 (high speed on software platforms), SNOW 3G (used in 3GPP).
  - Asymmetric ciphers: RSA (Rivest – Shamir – Adleman), ECC (Elliptic Curve Cryptography) - are mainly used to establish session keys (for example, in protocols like TLS) due to their high computational complexity.
  - Authentication and integrity algorithms: HMAC (Hash-based Message Authentication Code - based on SHA-2, SHA-3), AES-GCM (Galois / Counter Mode), AES-CCM (Counter with CBC-MAC) — modes that combine encryption and authentication.
  - Key and Pseudo-Random Sequence Generators: Deterministic RBGs (Random Bit Generators),

standards-compliant (NIST SP 800-90A), algorithms based on chaotic systems (for increased robustness).

- FPGA Implementation: Performance and Protection:
  - High Throughput: Hardware implementation of block ciphers (especially AES) on FPGAs provides orders of magnitude higher encryption/decryption speeds compared to software implementations on CPUs, easily reaching gigabit speeds [6, 7].
  - Parallelism and Pipelining: FPGAs allow efficient parallelization of processing of several data blocks or implementation of a deep pipeline for a single cipher. For example, AES in ECB mode (Electronic Codebook) can encrypt multiple blocks simultaneously. Pipelining each AES round allows multiple blocks to be processed at different stages of encryption simultaneously [3, 7].
  - Physical Security: FPGAs can provide better physical isolation of cryptographic cores from the rest of the system compared to software solutions on CPUs. There are FPGA design techniques to protect against side - channel attacks (Attacks, SCA), such as power consumption analysis (Power Analysis (DPA) or electromagnetic radiation analysis (EMA) [7].
  - Flexibility and Upgradeability: The FPGA's reconfigurability allows for cryptographic algorithms to be upgraded or replaced as standards change (e.g. from AES-128 to AES-256), vulnerabilities are discovered, or new threats (e.g. quantum) emerge.
  - Features and Limitations:
- Asymmetric cryptography (RSA, ECC) is extremely resource intensive on FPGAs, especially for long keys (2048+ bits for RSA, 256+ bits for ECC), and is often performed on dedicated coprocessors or in hybrid systems on CPUs.
- Effective implementation requires careful selection and tuning of block cipher modes (CBC, CTR, GCM).
- Secure generation, storage and management of cryptographic keys on FPGAs is a separate complex task that requires the use of protected memory areas (eFUSE, BBRAM) and hardware true random number generators (TRNGs).
- Implementation of complex algorithms (for example, AES with S-Box based on composite fields) also requires significant resources (logic, memory, DSP slices for operations in finite fields GF(2^8)) [6, 7].

III. Architectural Features and Design Solutions of SDR on FPGA

Implementation of SDR on FPGA imposes specific requirements and opens up unique opportunities that determine the system architecture:

- The Central Role of Parallelism and Pipelining: The ability of FPGAs to perform many operations simultaneously is their main advantage over sequential processors. This requires rethinking coding algorithms (especially FEC and compression) to identify and exploit the inherent parallelism. Pipelining operations allows achieving high clock rates and continuous data flow processing [3].
- Combating Resource Intensity: Coding algorithms, especially FEC (LDPC, Turbo) and complex video codecs (H.265), as well as cryptography, can consume the vast majority of FPGA resources:
  - Logical elements (LUTs, Flip-Flops): For the implementation of control logic, finite state machines, computing units.
  - Memory blocks (Block RAM, BRAM): For storing large parity check matrices (LDPC), transition tables (Viterbi), intermediate data of video codecs, data buffers.
  - DSP slices: Critical for performing multiply-accumulate (MAC) in FEC (Viterbi, Turbo codes), transforms (FFT, DCT) in source codecs, finite field operations in cryptography (AES).
  - Input/Output (I/O): For high-speed data exchange with ADC/DAC, memory, processor system.
  - Careful design, register-level optimization (RTL) and use of high-level synthesis (HLS) tools are necessary to achieve the required performance with available resources [3, 6].
- Flexibility and Reconfigurability: The Heart of SDR: The ability to dynamically reprogram the FPGA (on-the-fly or on reboot) to load different configurations (bitstreams) supporting different communication standards with their own unique modulation, demodulation and, most importantly, coding schemes (FEC, compression, encryption) is the key value of SDR [10].
- Hybrid Architectures: FPGA + CPU in SoC: Modern SDR platforms are increasingly built on the basis of systems-on-chip (SoC), combining programmable logic (FPGA) and powerful processor cores (ARM Cortex-A /R, x86) on a single chip (e.g. Xilinx Zynq Ultrascale+, Intel Agilex SoC). This allows for optimal distribution of coding tasks:
  - FPGA Fabric:
- Physical layer (PHY) in general: filtering, synchronization, correction.
- Modulation/Demodulation (QAM, OFDM, etc.).
- FEC Core: Encoding/Decoding (LDPC, Turbo, Viterbi, RS).
- Basic source encoding/decoding: Low latency audio/speech compression (ADPCM, CELP), video block pre-processing.
- Basic encryption/decryption of data streams: Implementation of symmetric ciphers (AES in GCM, CTR modes) at high speeds.
  - CPU (Processor System):
- Management of the protocol stack of higher levels (MAC, network , transport).
- Complex logic for establishing and maintaining communications, radio resource management.
- Key management and cryptographic protocols: Performing asymmetric cryptography, key exchange, authentication.
- High-level source codec management: Audio/video flow control, buffering, implementation of complex

parts of codecs (e.g. bitrate control in H.265), general-purpose data processing [6, 7].

- High-speed interfaces between FPGA and CPU (AXI, PCIe) provide efficient data exchange in such architecture.
- Latency: Real-Time Factor: Implementing latency-critical signal processing steps (demodulation, FEC decoding , basic encryption/decryption) directly on the FPGA ensures the lowest possible and predictable latency, which is vital for real-time applications (voice, control, AR/VR) [3, 9].
- Power efficiency: Hardware implementation of computationally intensive coding algorithms (FEC, cryptography, basic compression) on FPGAs often provides better performance per watt than implementations on CPUs or GPUs, especially in massively parallel computing [3].
- Development Complexity: Development and optimization of HDL code (VHDL/ Verilog) for complex coding algorithms requires high qualification and time. The use of high-level synthesis tools (High-Level Synthesis (HLS), which allows generating HDL code from C/C++ or Python descriptions , significantly speeds up development, especially for complex mathematical algorithms (LDPC, FFT for video codecs). However, to achieve maximum performance and minimize resource usage, manual optimization at the RTL level is often still required [3, 6].

IV. Practical Applicability: Examples and Industries

The theoretical advantages of SDR on FPGA with efficient coding are widely used in various industries:

- Fifth Generation Mobile Networks (5G NR):
  - FEC: FPGAs are the workhorse for implementing LDPC (for user data - PDSCH/PUSCH) and polar codes (for control information - PDCCH/PUCCH) decoders in base stations (gNodeB) and user equipment (UE), especially in early deployments and prototypes. The required gigabit per second speeds and low latency make FPGAs an ideal solution [1, 4].
  - Source Coding: Speech Codec Implementation (EVS - Enhanced Voice Services) with ultra-low latency for VoNR (Voice over New Radio).
  - Cryptography: Hardware-accelerated encryption (128/256-bit AES, SNOW 3G) and authentication (AES-GCM) to protect user traffic and signaling.
- Modern Wireless Local Area Networks (Wi-Fi 6/7 - IEEE 802.11ax/ be):
  - FEC: LDPC decoders are a must-have feature of high-end access points and client devices, providing high throughput and reliability in interference-ridden environments (especially in dense OFDMA/MU-MIMO scenarios) [5].
  - Cryptography: WPA3 acceleration (AES-GCMP, Simultaneous Authentication of Equals - SAE).
- Satellite Communications and Broadcasting (DVB-S2/S2X, DVB-T2):
  - FEC: Cascaded FEC schemes (BCH + LDPC in DVB-S2/S2X) require significant computational resources for decoding at high modulation rates (APSK-16/32). Implementation on FPGAs allows achieving the required performance in satellite modems and receivers [2, 9].
- Tactical and Critical Communication Systems:
  - FPGA SDR Flexibility: Allows a single platform to support multiple military and government communications standards (SATCOM, HF/VHF/UHF radio).
  - Source Coding: Implementation of low-latency speech codecs (MELP, AMBE) for digital tactical radios.
  - FEC: Using error-correcting codes (Reed-Solomon, convolutional, LDPC) in difficult signal propagation conditions.
  - Cryptographic stability: Mandatory hardware implementation of strong encryption algorithms (AES-256, stream ciphers) and authentication, often with increased protection measures against SCA [7, 10].
- Internet of Things (IoT) and Industrial Internet (IIoT):
  - Power efficiency: Use FPGAs in gateways or base stations to efficiently process FEC (e.g. LDPC in NB-IoT, LTE-M) and encrypt traffic from multiple sensors.

Hybrid SoCs: Allow integration of SDR and network/data management functions on a single chip.

V. Conclusions

The analysis conducted convincingly demonstrates that FPGAs are not just a preferred, but often an indispensable platform for implementing coding methods in modern high-performance SDR systems. This is due to the unique architectural features of FPGAs, primarily their ability to massively parallel data processing and provide low, deterministic latency.

1. Channel Coding (FEC): Certainly the most demanding class of algorithms for FPGA resources and at the same time the most critical for performance. The implementation of modern high-performance codes, such as LDPC (5G NR, Wi-Fi 6/7) and turbo codes (4G LTE), at gigabit speeds is possible only on FPGAs or ASICs. Their hardware implementation provides the necessary throughput, reliability of error correction and minimal delay, which are the basis for the operation of modern wireless communication standards [1, 2, 4, 5].
2. Source Coding: Efficiently implemented on FPGA primarily for latency-critical applications (voice communications, interactive video) and systems with extreme bandwidth or energy efficiency requirements (satellites). For complex codecs (H.265/HEVC), a hybrid approach is optimal , where the FPGA takes on resource-intensive calculations, and the CPU - flow control [3, 6, 7].
3. Crypto-Resistant Encryption: Hardware implement-tation of symmetric ciphers (especially AES) and authentication modes (AES-GCM) on FPGAs provides the highest throughput and security required to protect information in military, government, and commercial communications systems. Asymmetric cryptography is more often performed on SoC CPUs [6, 7].

The key trend is the widespread use of hybrid SoC platforms (FPGA + CPU), which provide an optimal distribution of tasks: hardware acceleration of the most resource-intensive and latency-critical coding functions (FEC, basic ciphers, low-latency compression) on the FPGA and execution of complex control, protocol stack and high-level control on the CPU. High-level synthesis (HLS) tools significantly accelerate the development of complex coding algorithms for FPGAs.

Development prospects are closely linked to support for new communication standards (6G, Wi-Fi 8), where the requirements for throughput, latency and FEC efficiency will only increase, as well as with the introduction of quantum-resistant cryptographic algorithms (PQC - Post-Quantum Cryptography), the computational complexity of which will also require hardware acceleration on FPGAs. The synergy of SDR and FPGAs will continue to be the basis for creating flexible, high-performance and secure wireless communication systems of the future.

REFERENCES

[1] Richardson , T., Kudekar , S. Design of Low-Density Parity Check Codes for 5G New Radio // IEEE Communications Magazine . 2018. Vol . 56, no . 3. P. 28–34. DOI: 10.1109/MCOM.2018.1700839.

[2] Lin , S., Costello , DJ Error Control Coding : Fundamentals and Applications . 2nd ed . Pearson Prentice Hall , 2004. 1260 p .

[3] Woods , R., McAllister , J., Lightbody , G., Yi , Y. FPGA-Based Implementation of Signal Processing Systems . 2nd ed . Wiley , 2017. 434 p . DOI: 10.1002/9781119077954.

[4] 3GPP TS 38.212:NR; Multiplexing and channel coding (Release 17). 3rd Generation Partnership Project , 2023. URL: https://www.3gpp.org/ftp/Specs/archive/38_series/38.212/ (accessed: 07.08.2025).

[5] IEEE Standard for Information technology - Telecommunications and information exchange between systems - local and metropolitan area networks — specific requirements . Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN. IEEE Std 802.11ax-2021. IEEE, 2021. URL: https://standards.ieee.org/ieee/802.11ax/7339/ (accessed: 07.08.2025).

[6] Xilinx . Versal ACAP AI Engine Programming Environment User Guide (UG1076). v2023.2, 2023. URL: https://docs.xilinx.com/r/en-US/ug1076-ai-engine-environment/ (accessed: 07.08.2025).

[7] Intel . Intel® Agilex ™ FPGA and SoC FPGA Development Kit User Guide . UG-20270, 2023. URL: https://www.intel.com/content/www/us/en/docs/programmable/683472/current/overview.html (accessed: 07.08.2025).

[8] Berrou , C., Glavieux , A., Thitimajshima , P. Near Shannon Limit Error-Correcting Coding and Decoding : Turbo-Codes (1) // Proceedings of ICC '93. IEEE, 1993. Vol . 2. P. 1064–1070. DOI: 10.1109/ICC.1993.397441.

[9] Sklar , B. Digital Communications : Fundamentals and Applications . 2nd ed . Prentice Hall PTR, 2001. 1079 p .

[10] Tuttlebee , W. (Ed .). Software Defined Radio : Enabling Technologies . Wiley , 2002. 386 p . DOI: 10.1002/0470841899.