IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 22, 2020, accepted July 2, 2020, date of publication July 8, 2020, date of current version July 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007923

An Efficient FEC Encoder Core for VCM LEO Satellite-Ground Communications

JING KANG[1,2], JUN SHE AN[1], AND BINGBING WANG[1,2]

[1]Key Laboratory of Electronics and Information Technology for Space Systems, National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China

[2]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100190, China

Corresponding author: Jing Kang (k_naive@163.com)

This work was supported by the Space Science Leading Satellite Project under Project XDA15320100.

ABSTRACT A powerful forward error correction (FEC) scheme based on the serial concatenation of Bose-Chaudhuri-Hocquenghen (BCH) and low-density parity-check (LDPC) codes has been adopted by the second generation digital video broadcast (DVB-S2) standard due to their near Shannon limit performance. This paper proposes an efficient FEC encoder core to support different DVB-S2 codes for variable coding modulation (VCM) schemes of low earth orbit (LEO) satellite-ground communications. By exploring the properties of different concatenated BCH-LDPC codes and reusing the computation units and memories, the compatibility is achieved and the utilization of hardware resources is improved. Besides, parallel computing is performed to speed up the acquisition of check bits, which allows the encoding throughput to be increased. Hardware architecture implementation on Xilinx XC7K325t field programmable gate array (FPGA) shows that the encoding throughput rate of the proposed FEC encoder core can reach 1.19 Gb/s. And the overall data throughput is improved by 30.9% compared with the constant coding modulation (CCM) schemes.

INDEX TERMS FEC encoder, LDPC, BCH, VCM, satellite-ground communications, FPGA.

I. INTRODUCTION

With relative motion between low earth orbit (LEO) satellite and ground station, the communication link budget varies in the transmission window. Conventional transmitting systems with constant coding modulation (CCM) schemes are normally designed based on the worst link budget (maximum distance and worst environmental conditions) to guarantee a desired transmission quality. However, this results in the waste of link resources [1]. To improve the efficiency of satellite-ground communications, variable coding modulation (VCM) schemes have been proposed and studied [2]–[5].

The second generation digital video broadcast (DVB-S2) standard [5] has adopted a powerful forward error correction (FEC) code based on the serial concatenation of Bose-Chaudhuri-Hocquenghen (BCH) and low-density parity-check (LDPC) codes for VCM transmitting systems. This new FEC structure, coupled with high order modulations, can provide capacity gains of about 30% over the previous digital video broadcast (DVB-S) standard [6], where LDPC codes play a fundamental role in the performance improvement. Nevertheless, the high-speed requirements, long block lengths and adaptive encoding defined in each VCM mode present complex challenges on the implementation of an encoder fully compliant with all code configurations. On the other hand, the encoder is also required to control data flow properly when VCM modes change.

Implementations of the FEC encoders based on the DVB-S2 standard can be found in [7]–[14]. In [7], [8], silicon implementations of DVB-S2 compliant codec based on 64800b LDPC and BCH codes were present, with a maximum throughput of 135Mb/s. Reference [9] proposed a serial BCH encoder and a parallel LDPC encoder, both of which support 21 different DVB-S2 code configurations, but they are not cascaded for FEC system-level design and validation. Reference [10] designed a BCH-LDPC cascade encoder that supports a single code length 64800 and code rate 1/2, but the data throughput is limited due to its serial-in and serial-out hardware architecture, which cannot meet the requirement of high-speed applications. In [11], parallel implementation of BCH and LDPC encoders was proposed and static random access memory (SRAM) was employed to store check bits, therefore both row and column addresses are required to

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Huo.

search parity-check bits corresponding to the parallel information bits, which increases the complexity of address management and the controller. References [12], [13] developed efficient LDPC encoders for satellite ground terminals, but large size barrel shifter registers employed in the design increased the power consumption. Reference [14] used a compacted memory mapping of the parity-check equations and performed partial-parallel computation of the parity-check bits to achieve a high throughput LDPC encoder, but the parallel output of the encoder is not in natural order and an interleaver design need to be included, which increases the hardware complexity.

In this paper, we propose an efficient FEC encoder core for VCM schemes of LEO satellite-ground data transmissions. The contributions are as follows:

1) Considering the periodicity and similarity of DVB-S2 LDPC codes, we propose a fast accumulate semi-parallel recursive (FASPR) LDPC encoding algorithm for efficient hardware implementation, which can compute parallel parity-check bits at a time in natural order.
2) The powerful control modules are designed that can reuse computation units and memories efficiently. So that the proposed FEC encoder core is compatible with various VCM modes and reduces hardware resources consumption, and correctness is guaranteed when VCM modes change.
3) Parallel computation of check bits is performed to achieve high throughput.

The rest of this paper is organized as follows: Section II describes the parallel BCH encoding and the FASPR LDPC encoding algorithm. Section III makes a detail description of the proposed FEC encoder core. The implementation results are reported in section IV. Finally, section V concludes the paper.

II. ALGORITHMS OF FEC ENCODING

A. PARALLEL BCH ENCODING

BCH codes are a class of cyclic codes with powerful error-correcting capability that widely used in modern communication systems [15], [16].

The encoding process of a systematic (n, k) BCH code is as follows:

Step1: Multiply the information polynomial $m(x) = m_{k-1}x^{k-1}+m_{k-2}x^{k-2}+\ldots+m_0$ by x^{n-k}. This multiplication is equivalent to shifting the information bits by $(n-k)$ positions to the right.

Step 2: Obtain the generator polynomial $g(x)$ of the t-error correcting BCH code by $g(x) = LCM(g_1(x), \cdots, g_t(x))$, where $g_i(x)$ is the minimum polynomial of $g(x)$ and LCM represents the least common multiple of these polynomials.

Step 3: Divide $m(x) \cdot x^{n-k}$ by $g(x)$, and let $r(x) = r_{n-k-1}x^{n-k-1} + \ldots + r_1x + r_0$ be the remainder. The coefficients of the remainder stand for the check bits.

Step 4: Add the computed remainder to $m(x)\cdot x^{n-k}$ to obtain the codeword polynomial $c(x) = m(x) \cdot x^{n-k} + r(x)$.

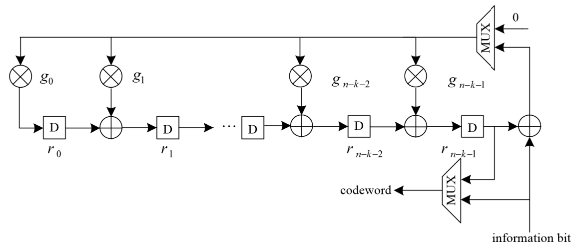

FIGURE 1. Basic LFSR architecture.

A systematic binary (n, k) BCH code can be implemented by the serial linear feedback shift register (LFSR) architecture illustrated in Fig.1. The symbol $\oplus$ indicates an *xor* operation, and the multiplication $\otimes$ can be simplified to a connection or disconnection when g_i is "1" or "0". Registers $r_0, r_1, \cdots, r_{n-k-1}$ represent the coefficients of the remainder.

During the first k clock cycles, k information bits are imported to the LFSR serially with the most significant bit (MSB) first. Meanwhile, the information bits are also sent to the output port to form the systematic part of the codeword. After the k-*th* clock cycle, the LFSR contains $(n-k)$ coefficients of $r(x)$. They are then shifted out of the registers to form the remaining bits of the codeword.

While the encoding scheme mentioned above cannot meet the requirement of high-speed applications due to the serial-in and serial-out architecture. Herein, we propose a parallel BCH encoding algorithm based on state-space computation proposed in [17].

Assuming that a $(n-k)$ dimensional state vector $\boldsymbol{R}(t) = [r_{n-k-1}(t), r_{n-k-2}(t) \cdots r_0(t)]^T$, where $r_i(t)$ represents the state of the i-*th* remainder register at moment t, then $\boldsymbol{R}(t+1)$ can be expressed as:

$$\boldsymbol{R}(t+1) = \boldsymbol{F} \cdot (\boldsymbol{R}(t) + \boldsymbol{M}(t+1)) = \boldsymbol{F} \cdot \boldsymbol{R}(t) + m_{t+1} \cdot \boldsymbol{G}. \tag{1}$$

where matrix

$$\boldsymbol{F} = \left[\boldsymbol{G} | \tfrac{\boldsymbol{I}}{0}\right] = \begin{bmatrix} g_{n-k-1} & 1 & 0 & \cdots & 0 \\ g_{n-k-2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g_1 & 0 & 0 & \cdots & 1 \\ g_0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and $\boldsymbol{M}(t+1) = [m_{t+1} | 0, \cdots, 0]^T$ is a $(n-k)$ dimensional vector with the first element equals to the $(t+1)$-*th* information bit m_{t+1} and the others are zeros.

According to (1), $\boldsymbol{R}(t+D)$ can be derived from recursive deductions as:

$$\boldsymbol{R}(t+D) = \boldsymbol{F}^D \cdot (\boldsymbol{R}(t) + \boldsymbol{M}(t+D)). \tag{2}$$

where $\boldsymbol{M}(t+D) = [m_{t+1}, m_{t+2}, \cdots, m_{t+D} | 0, \cdots, 0]^T$ is the vector of D information bits, and

$$\boldsymbol{F}^D = \left[\boldsymbol{F}^{D-1} \cdot \boldsymbol{G} \middle| \cdots \middle| \boldsymbol{F} \cdot \boldsymbol{G} \middle| \begin{matrix} \text{the first}(n- \\ k-D+1) \\ \text{columns of} \boldsymbol{F} \end{matrix} \right]. \quad (3)$$

Equation (2) means that $\boldsymbol{R}(t+D)$ can be obtained from $\boldsymbol{R}(t)$ directly, and the "1"s in the matrix $\boldsymbol{F}^D$ represent *xor* operations between the remainder registers and the vector of D information bits. By using the parallel BCH encoding algorithm, D information bits can be processed simultaneously by the encoder, therefore only k/D clock cycles are required to obtain the state vector of check bits, which increases the encoding throughput by D times.

B. THE PROPOSED FASPR LDPC ENCODING

The DVB-S2 standard has adopted irregular repeat accumulate (IRA)-LDPC codes, which has the linear encoding complexity [18], [19]. IRA-LDPC codes are normally characterized by a parity check matrix $\boldsymbol{H}$:

$$\begin{aligned} &\boldsymbol{H}_{(n-k)\times n} \\ &= [\boldsymbol{A}_{(n-k)\times k} \mid \boldsymbol{B}_{(n-k)\times(n-k)}] \\ &= \left[\begin{array}{ccc|cccc} a_{00} & \cdots & a_{0,k-1} & 1 & 0 & \cdots & 0 \\ a_{10} & \cdots & a_{1,k-1} & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \ddots & \vdots \\ a_{n-k-2,0} & \cdots & a_{n-k-2,n-k-1} & 0 & 0 & 1 & 0 \\ a_{n-k-1,0} & \cdots & a_{n-k-1,n-k-1} & 0 & 0 & 0 & 1 \end{array}\right] \end{aligned} \quad (4)$$

where $\boldsymbol{B}$ is a staircase lower triangular matrix and $\boldsymbol{A}$ is a sparse matrix.

The codes defined by (4) are systematic, i.e., the codeword has the following format $\boldsymbol{C} = [\boldsymbol{I} \mid \boldsymbol{P}]$, with the information bits $\boldsymbol{I} = [i_0, i_1, \cdots, i_{k-2}, i_{k-1}]$ being associated to the sparse matrix $\boldsymbol{A}$ and the parity-check bits $\boldsymbol{P} = [p_0, p_1, \cdots, p_{n-k-1}]$ being associated to the matrix $\boldsymbol{B}$.

Due to the bi-diagonal structure of matrix $\boldsymbol{B}$, parity-check bits can be calculated recursively:

$$\begin{cases} p_0 = a_{00}i_0 \oplus a_{01}i_1 \cdots \oplus a_{0,k-1}i_{k-1}, \\ p_1 = a_{10}i_0 \oplus a_{11}i_1 \cdots \oplus a_{1,k-1}i_{k-1} \oplus p_0, \\ \vdots \\ p_{n-k-1} = a_{n-k-1,0}i_0 \oplus a_{n-k-1,1}i_1 \cdots \oplus p_{n-k-2} \end{cases} \quad (5)$$

where $\oplus$ is the GF(2) addition operator.

The design technique of matrix $\boldsymbol{A}$ is based on some periodicity constraints, which divide the information nodes into groups of 360 consecutive ones. All the information nodes of a group have the same weight v, and indices of parity-check nodes that connected to the j-$th(1 \le j \le 360)$ information node of the group can be obtained by:

$$\{d_i + (j-1) \cdot q\} \bmod (n-k), \quad i \in \{1, 2, \ldots, v\}. \quad (6)$$

where d_i is the indices of the parity-check nodes that connected to the first information node of the group and $q = (n-k)/360$.

By computing (5) and considering the periodicity and similarity of DVB-S2 IRA-LDPC codes, we propose the FASPR LDPC encoding algorithm.

The indices of the information bits that participate in the parity-check node restriction r can be denoted by IN(r), and the indices of parity-check bits connected to the information node c can be denoted by CN(c). Therefore, (5) can be rewritten as:

$$\begin{cases} p_0 = \bigoplus_{z \in \mathrm{IN}(0)} i_z = S_0, \\ p_1 = \bigoplus_{z \in \mathrm{IN}(1)} i_z \oplus p_0 = S_1 \oplus S_0, \\ \vdots \\ p_r = S_r \oplus p_{r-1} = \bigoplus_{i=0}^{r} S_i. \end{cases} \quad (7)$$

where $S_r = \bigoplus_{z \in IN(r)} i_z,\ r \in \{0, 1, \cdots, n-k-1\}$.

For each new parallel D information bits received by the encoder, the associated S_r can be updated according to:

$$\begin{aligned} &\textit{for } c = 0 : D : \text{k-D} \\ &\quad \textit{for } r \in CN(c) \textit{ do} \\ &\quad\quad S_r = S_r \oplus i_c \\ &\quad\quad S_{(r+q)\bmod(n-k)} = S_{(r+q)\bmod(n-k)} \oplus i_{c+1} \\ &\quad\quad \vdots \\ &\quad\quad S_{r+(D-1)q\bmod(n-k)} = S_{r+(D-1)q\bmod(n-k)} \oplus i_{c+D-1} \\ &\quad \textit{end} \\ &\textit{end} \end{aligned} \quad (8)$$

We can simplify S_r by:

$$S_r = \begin{cases} S_r & (i_c = 0) \\ \overline{S_r} & (i_c = 1) \end{cases} \quad (9)$$

where $\overline{S_r}$ is the bit inversion value of S_r. It indicates that S_r is updated only when the corresponding information bit is "1". And the addition operation is replaced by the inverse operation, which reduces the hardware overhead.

Once S_r values are known, the parity-check bits can be obtained by (7). S_r can be stored compactly by using the following binary matrix:

$$\boldsymbol{S} = \begin{bmatrix} S_0 & S_1 & \cdots & S_{D-1} \\ S_D & S_{D+1} & \cdots & S_{2D-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{(L-1)D} & S_{(L-1)D+1} & \cdots & S_{LD-1} \end{bmatrix}_{L \times D} \quad (10)$$

where $L = (n-k)/D$. Therefore, D parity-check bits can be accumulated at a time in natural order by using the procedure (11), as shown at the bottom of the next page.

The proposed FASPR LDPC encoding algorithm explores the property of the adopted codes and uses a compact memory to store the simplified S_r. And by performing D parallel

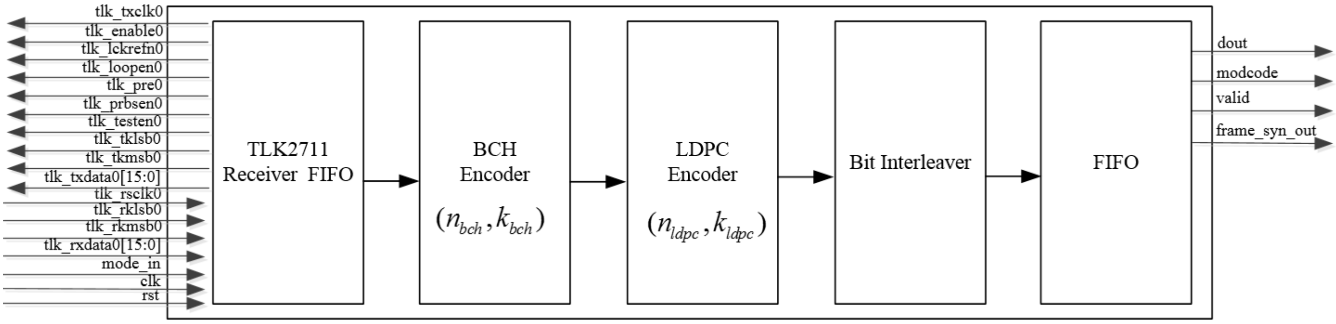

FIGURE 2. Block diagram of the FEC encoder core.

parity-check bits at a time in natural order, low encoding latency is achieved.

III. ARCHITECTURE OF THE PROPOSED FEC ENCODER CORE

A. OVERALL ARCHITECTURE

It is well known that propagation conditions keep changing during the LEO satellite pass above a ground station. When the distance decreases with higher elevation, the free space loss (FSL) decreases and the link budget is improved [20]. The dynamic link budget allows the use of variable modulations and code rates, which increase the download throughput without additional power consumption.

Herein, we propose a FEC encoder core for VCM schemes based on algorithms described in section II. Fig.2 shows a block diagram of the main blocks of the FEC encoder core.

The proposed FEC encoder core consists of the outer BCH encoder, inner LDPC encoder, and the bit interleaver, supporting all DVB-S2 code configurations. The frames to be encoded are input to TLK2711 multigigabit transceivers, and the encoded frames are output as symbols for direct connection to the mapper. Besides, the FEC encoder core computes the check bits in a parallel way, and parallel degree D is determined by modulation order, e.g., $D = 3$ for 8 phase shift keying (8PSK) and $D = 4$ for 16 amplitude phase shift keying (16APSK).

B. BCH ENCODER

In the DVB-S2 standard, BCH codes have the capability to correct $t = \{8, 10, 12\}$ errors and have 4 different lengths of generator polynomial g(x) for different code rates and frame lengths. Table 1 shows the correcting capabilities and generator polynomials supported by DVB-S2 standard. As shown in Table 1, the degree parameter represents the highest power of the generator polynomial g(x). And it is noteworthy that the coefficients of g(x) is expressed in hexadecimal, and the rightmost bit is padded with zeros to allow a hexadecimal number.

TABLE 1. Correcting capabilities and generator polynomials supported by DVB-S2 standard.

	code rate	t	coefficients of g(x)	degree
Normal frame (64800bit)	1/4, 1/3, 2/5, 1/2, 3/5, 3/4, 4/5, 5/6	12	A7130741C22E288 E2867966C6E1A84 4481A3C2FBB3012 AF38	192
	2/3, 5/6	10	B00A8676FE15198 FB53C2B81F7E891 80DC5DB2C88	160
	8/9, 9/10	8	8E0392AFB893CBD E8CFE36BA827CB3 158	128
Short frame (16200bit)	1/4, 1/3, 2/5, 1 /2, 3/5, 2/3, 3 /4, 4/5, 5/6, 8/9	12	A0316DF54C34D93 16691D1C834A947 F3EBE88C82D28	168

The generator polynomial, i.e., the matrix $\boldsymbol{F}^D$, need to be changed to accommodate the correcting capability of different VCM modes, which leads to change in the hardware architecture. To implement the reconfigurable parallel BCH encoder with variable correcting capacities, we use MATLAB to precompute $\boldsymbol{F}^D$ required by each VCM mode. According to (2), the "1"s in the matrix $\boldsymbol{F}^D$ indicate *xor* operations

$$\begin{cases} \left[S_0 \ \bigoplus_{r=0}^{1} S_r \ \cdots \ \bigoplus_{r=0}^{D-1} S_r\right] = \left[p_0 \ p_1 \ \cdots \ p_{D-1}\right], \\ D'b\{p_{D-1}\} \oplus \left[S_D \ \bigoplus_{r=D}^{D+1} S_r \ \cdots \ \bigoplus_{r=D}^{2D-1} S_r\right] = \left[p_D \ p_{D+1} \ \cdots \ p_{2D-1}\right], \\ \vdots \\ D'b\{p_{(L-1)D-1}\} \oplus \left[S_{(L-1)D} \ \bigoplus_{r=(L-1)D}^{(L-1)D+1} S_r \ \cdots \ \bigoplus_{r=(L-1)D}^{LD-1} S_r\right] = \left[p_{(L-1)D} \ p_{(L-1)D+1} \ \cdots \ p_{LD-1}\right]. \end{cases} \tag{11}$$

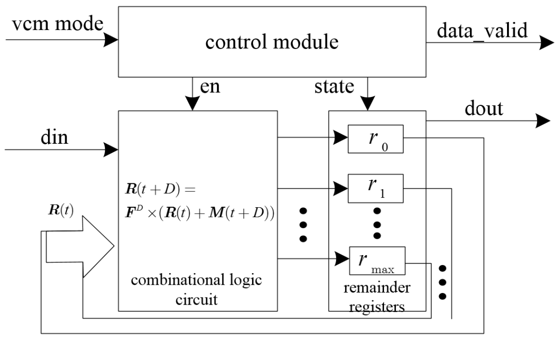

FIGURE 3. Architecture of reconfigurable parallel BCH encoder.

between remainder registers and information vector. Therefore, the BCH encoder can be easily implemented by a combinational logic circuit. Besides, zeros are padded in the most significant bits of the information vector to ensure that it is divisible by D. It is worth noting that the dimension of the matrix $\boldsymbol{F}$, the length of the remainder registers and the dimension of $\boldsymbol{G}$ need to be designed in the worst case, i.e., equal to the maximum degree of $g(x)$. Fig.3 shows the reconfigurable parallel BCH encoder architecture.

During the first $ceil(k_{bch}/D)$ clock cycles, the combinational logic circuit operates D parallel information bits with the state vector $\boldsymbol{R}(t)$ and sends the results $\boldsymbol{R}(t+D)$ back to the remainder registers. Meanwhile, the information bits are sent to the output port to form the systematic part of the codeword. After the $ceil(k_{bch}/D)$-th clock cycle, the remainder registers contain the state vector of check bits, they are then shifted out of the registers to form the remaining bits of the codeword.

C. LDPC ENCODER

Fig.4 shows the architecture of the proposed LDPC encoder, which composed of four major modules. The address computation module is responsible for computing the addresses and offsets of parity-check bits that corresponding to the information bits (This process can be denoted as Task 1.) The S_r module is responsible for storing and updating the values of S_r (Using Task 2 to represent this process.) The output process module is responsible for computing the parity-check bits (Expressed briefly as Task 3) and exporting the encoded codeword. The control module is designed for controlling the data flow of various VCM modes.

Dual-port random access memory (RAM) is employed to store and update S_r. To reduce encoding latency, a distributed storage strategy is used, which means $v \cdot D$ S_r RAMs are employed to perform (8). To make the encoder compliant with all VCM modes, $v_{\max} \cdot D_{\max}$ S_r RAMs with $(n-k)_{\max}/D_{\min}$ depth and $D_{\max}$ width are needed. The subscript *max* and *min* represent the maximum and minimum value of the variables, respectively. Besides, read-only memory (ROM) is employed to store the initial addresses of parity-check bits, which can be easily obtained from the annexes B and C of DVB-S2 standard tables [5]. Furthermore, since the hierarchical storage management is used, the appropriate row of the ROM should be selected based on the VCM mode.

The control module consists of state machines and control signals, which are used to dynamically reconfigure the encoder and to control the data flow. Fig.5 shows the state machines diagram and the encoding process.

To perform Task1, address and offset of the parity-check bit corresponding to each bit of D parallel information bits is

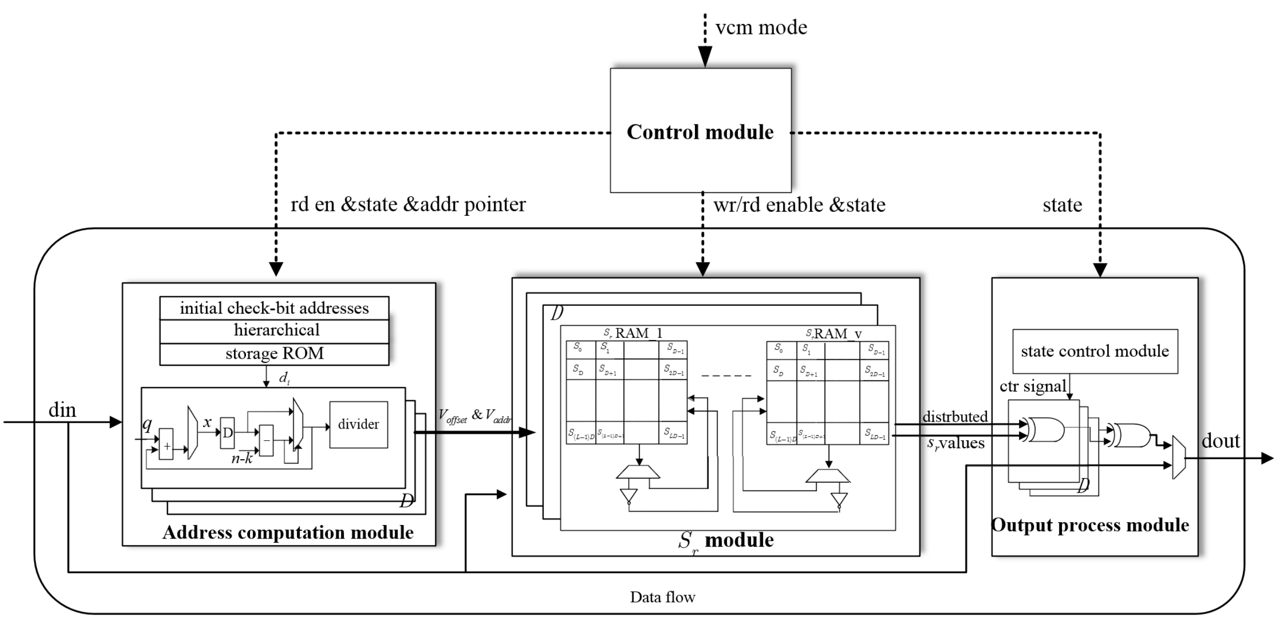

FIGURE 4. Architecture of LDPC encoder.

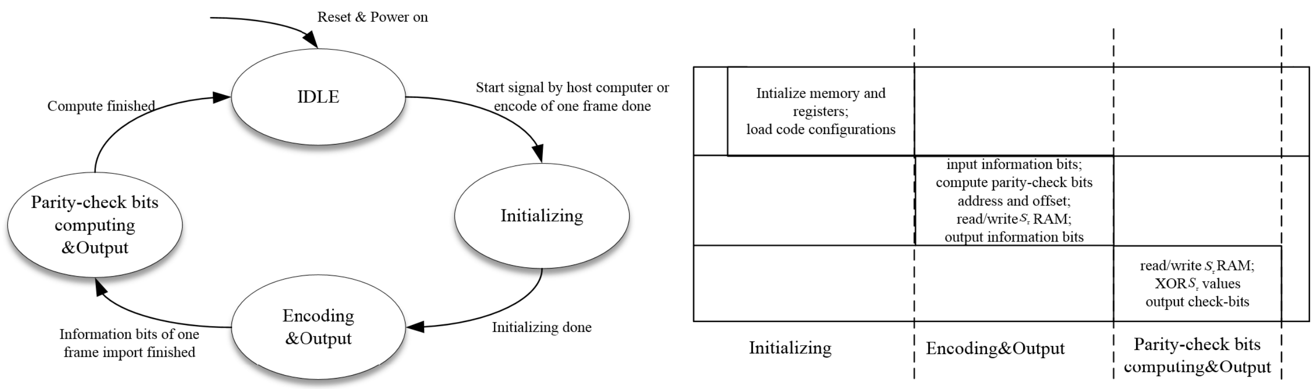

FIGURE 5. State machines diagram and encoding process of LDPC encoder.

computed respectively according to:

$$V_{addr} = \text{floor}(x/D), \tag{12}$$
$$V_{offset} = \text{mod}(x, D). \tag{13}$$

where x is parity-check bit address computed by (6).

Task 2 is performed simultaneously with Task 1. Based on the operating VCM mode, $v \cdot D$ S_r RAMs are enabled. The read enable signal of the RAM is valid only when the information bit is "1", and the write enable signal is one clock delay of the read enable signal. Controlled by the read/write enable signals, values in the addresses and offsets computed by Task 1 are read out, and inversed values are written back to the same addresses and offsets. Meanwhile, the information bits are also sent to the output port.

After all the information bits are imported to the encoder, the distributed S_r values are read out and processed by xor operation according to (11). Meanwhile, zero values should be written to the S_r RAMs to ensure that the subsequent data frames can be correctly encoded. Therefore, Task 3 is realized, obtaining D parallel parity-check bits per clock cycle in natural order. Compared with [14], which parallel output is not in natural order, and an interleaver design need to be included, our proposed architecture is more efficient and hardware friendly.

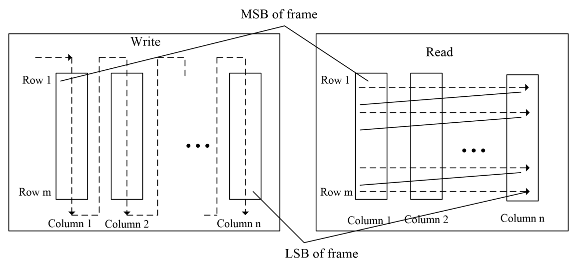

FIGURE 6. Bit interleaving scheme.

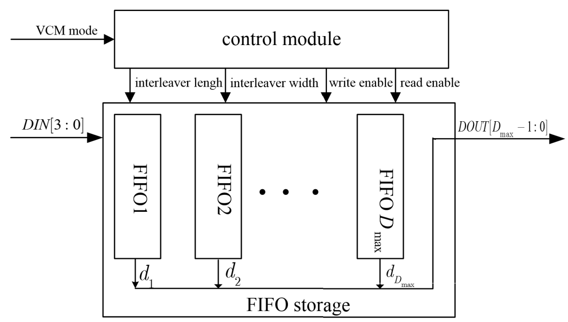

FIGURE 7. Architecture of parallel bit interleaver.

D. BIT INTERLEAVER

Channel interleaving is employed in most modern wireless communication systems to protect against burst error. Encoded frames are written into the interleaver column-wise, and read out row-wise to break the temporal correlation between successive bits, the process is depicted in Fig.6.

Fig.7 shows the architecture of the proposed parallel bit interleaver, with output as symbols that can be directly connected to the mapper.

The specific implementation steps of the parallel bit interleaver are as follows:

Step 1. To make the interleaver compliant with all VCM modes, $D_{\max}$ FIFOs are employed to store data. The size of each FIFO is $n_{\max}/D_{\min}$, where $n_{\max}$ is the maximum encoded frame length supported by the standard and $D_{\min}$ is the minimum parallel degree. The write width is 4 and the read width is 1 of each FIFO.

Step 2. The encoded frame is sequentially written into D FIFOs, i.e., the first n/D bits of the frame are written in the first FIFO and the subsequent n/D bits are written in the second FIFO, and so on. The write operation is completed when all bits of one encoded frame are written into D FIFOs. It is worth noting that n and D are the encoded frame length and parallel degree for current VCM mode.

Step 3. Once the write operation is completed, D bits data are read from D FIFOs, i.e., read one bit from each FIFO to form the parallel D bits data. The output port DOUT of the interleaver has width $D_{\max}$ and provides output as symbols.

TABLE 2. DOUT ordering for each modulation type.

	8PSK	16APSK	32APSK
DOUT(4)	$\boldsymbol{c}(3i+2)$	$\boldsymbol{c}(4i+3)$	$\boldsymbol{c}(5i+4)$
DOUT(3)	$\boldsymbol{c}(3i+1)$	$\boldsymbol{c}(4i+2)$	$\boldsymbol{c}(5i+3)$
DOUT(2)	$\boldsymbol{c}(3i)$	$\boldsymbol{c}(4i+1)$	$\boldsymbol{c}(5i+2)$
DOUT(1)	logic 0	$\boldsymbol{c}(4i)$	$\boldsymbol{c}(5i+1)$
DOUT(0)	logic 0	logic 0	$\boldsymbol{c}(5i)$

For $D<D_{\max}$, pad zeros in the least significant bits (LSB) of DOUT. The ordering of DOUT bits for each modulation type is summarized in Table 2. As shown in Table 2, for example, the symbol output needs only 4 bits when 16APSK is adopted, which means that DOUT (4:1) provides $\boldsymbol{c}(4i+3)$, $\boldsymbol{c}(4i+2)$, $\boldsymbol{c}(4i+1)$, and $\boldsymbol{c}(4i)$ of the encoded frame bits, and DOUT(0) is logic 0. It is noted that $\boldsymbol{c} = [\boldsymbol{c}_0, \boldsymbol{c}_1 \cdots \boldsymbol{c}_{n-2}, \boldsymbol{c}_{n-1}]$ is the encoded frame vector and $0 \le i \le n/D$.

IV. FPGA IMPLEMENTATION AND ANALYSIS

The proposed FEC encoder core is synthesized and implemented on the Xilinx XC7K325t FPGA for a practical LEO satellite-ground data transmission scenario. In addition, it is known that SRAM-based FPGAs suffer from single event effect in radiation environments, therefore we combine scrubbing and triple modular redundancy (TMR) technique to substantially lower the faults in time and increase the reliability of our design. The VCM schedule is shown in Fig.8. As shown in Fig.8, three VCM modes are adopted during the transmission window, when elevation angle is greater than 5-degree and less than 15-degree, VCM mod1 is adopted with frame length $n = 16200$, code rate $R = 2/3$, and modulation type 8PSK. When elevation angle is greater than 15-degree and less than 25-degree, VCM mod2 is adopted with frame length $n = 16200$, code rate $R = 2/3$, and modulation type 16APSK. And when elevation angle is greater than 25-degree VCM mod3 is adopted with frame length $n = 16200$, code rate $R = 4/5$, and modulation type 16APSK.

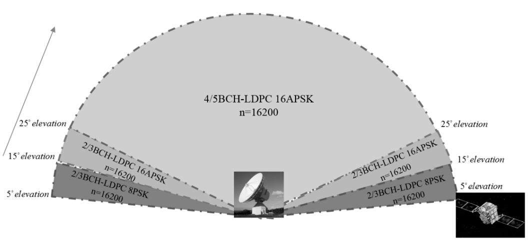

FIGURE 8. VCM schedule of one LEO satellite-ground data transmission scenario.

A. SIMULATION RESULTS

Fig.9 shows the I/O interface of the top FEC encoder core module VCM_encoder. And the detailed descriptions of the I/O ports of the top module are given in Table 3. Fig.10 shows the simulation results of the FEC encoder core.

In Fig.10, with the *mode_in* signal changes during the simulation time, the proposed FEC encoder core is dynamically

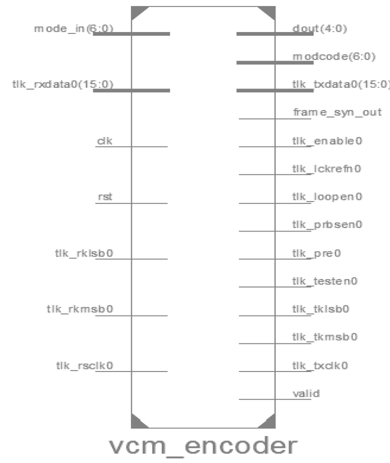

FIGURE 9. I/O interface of the VCM_encoder.

TABLE 3. I/O ports description of top module.

Port name	Port type	Description
clk	input	Clk signal
rst	input	Reset signal
mode_in[6:0]	input	VCM mode input port: parameters to be used to encode frame (frame length, code rate,modulation type)
tlk_rsclk0	input	Recovered clock
tlk_rklsb0	input	K-code indicator/PRBS test results
tlk_rkmsb0	input	K-code indicator.
tlk_rxdata0[15:0]	input	Receive data bus: information to be encoded
tlk_txclk0	output	Reference clock.
tlk_enable0	output	Device enable.
tlk_lckrefn0	output	Lock to reference.
tlk_loopen0	output	Loop enable.
tlk_pre0	output	Preemphasis control.
tlk_prbsen0	output	PRBS test enable.
tlk_testen0	output	Test mode enable.
tlk_tklsb0	output	K-code generator(LSB).
tlk_tkmsb0	output	K-code generator(MSB).
tlk_txdata0[15:0]	output	Transmit data bus.
dout[4:0]	output	Output data: encoded frame output
modcode[6:0]	output	VCM mode output port: parameters identification for current output frame
frame_syn_out	output	Frame synchronization signal
valid	output	Data valid : set high when data is valid

reconstructed and operated. The uncoded information bits are imported to the FEC encoder core via *tlk_rxdata*0 port through TLK2711 multigigabit transceivers. After BCH encoding, LDPC encoding, and bit interleaving stages, the encoded frame is output on *dout* as symbols, along with parameter identification *modcode* for current output frame. The *valid* signal is asserted to indicate that data on *dout* is valid. The frame synchronization signal *frame_syn_out* is asserted for a clock cycle when the first item of the encoded frame is presented on *dout*.

B. HARDWARE UTILIZATION

Table 4 presents hardware utilization of the proposed FEC encoder modules and comparison of this work with other state-of-art designs in [9], [12].

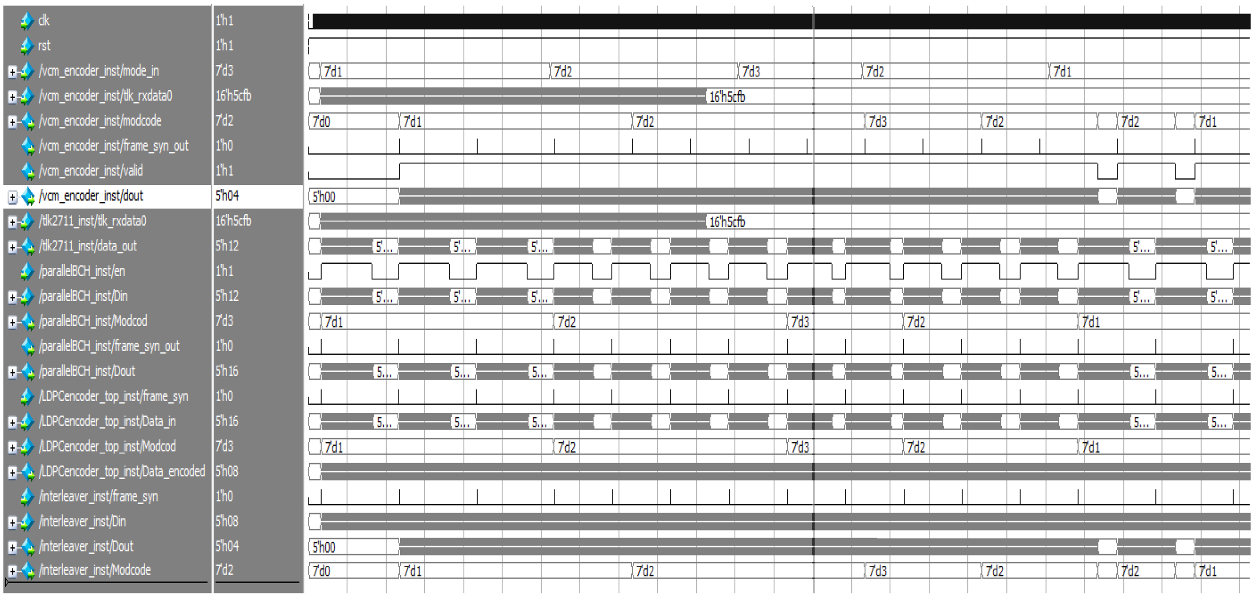

FIGURE 10. Simulation results of the FEC Encoder core.

TABLE 4. Hardware utilization and comparison of this work with existing designs.

	BCH encoder		LDPC encoder			FEC encoder core
	proposed	[9]	proposed	[12]	[9]	proposed
FPGA	XC7K325t	XC2VP30	XC7K325t	XC6VLX240T	XC2VP30	XC7K325t
Slices	-	548(4%)	-	-	4383(32%)	-
Flip-flops	-	548(2%)	-	3400(1%)	822(3%)	-
Registers	545(0.1%)	-	4578(1%)	3070(2%)	-	7720(2%)
LUTs	646(0.3%)	1096(4%)	6618(3%)	-	-	8884(4%)
BRAMs	-	-	34(7%)	20(5%)	28(20%)	49(11%)

As shown in Table 4, the proposed FEC encoder core occupies slice registers at 2%, slice look-up-tables (LUTs) at 4%, block RAMs (BRAMs) at 11%. Compared with the serial BCH encoder described in [9], which supports only one mode, our proposed parallel BCH encoder can support three VCM modes and reduce the LUTs consumption by 41%. The BRAMs requirements increase due to the distributed storage strategy employed in our design. As the proposed LDPC encoder is compatible with three VCM modes, the overall hardware complexity increases slightly compared with the LDPC encoder in [9], [12], which supports a single mode. It shows that the proposed FEC encoder core reaches an excellent compromise between flexibility and hardware utilization.

C. PERFORMANCE ANALYSIS

The timing analysis results show that it is possible to operate the FEC encoder core at a maximum clock frequency f_{clk}=389.5MHz, the encoding throughput rate for each VCM mode is given by:

$$T = \frac{n \cdot \eta \cdot f_{clk}}{C_p}. \tag{14}$$

where n is the encoded frame length, $\eta = k_{bch}/n$ is the efficiency of the FEC encoder core, and C_p is the number of clock cycles that it takes to process a frame at a particular VCM mode. C_p can be calculated by:

$$C_p = n/D + c_{delay}. \tag{15}$$

where D is parallel degree determined by modulation order, c_{delay} is a constant delay equals to 26 for initializing and configuring.

For VCM satellite-ground data transmission systems, the overall data throughput T_{VCM}, i.e., the total amount of data transmitted by the satellite to the ground during the transmission window, can be calculated as:

$$T_{VCM} = \sum_i T_{modi} \cdot \Delta t_{modi}. \tag{16}$$

where T_{modi} and Δt_{modi} is the throughput rate and the lasting time for each VCM mode.

According to the STK simulation results, the transmission window lasts 1080s, with Δt_{mod1}= 380s, Δt_{mod2} = 280s, Δt_{mod3} = 420s. Fig.11 summarizes the throughput rate of each VCM mode and the overall data throughput of VCM and CCM systems.

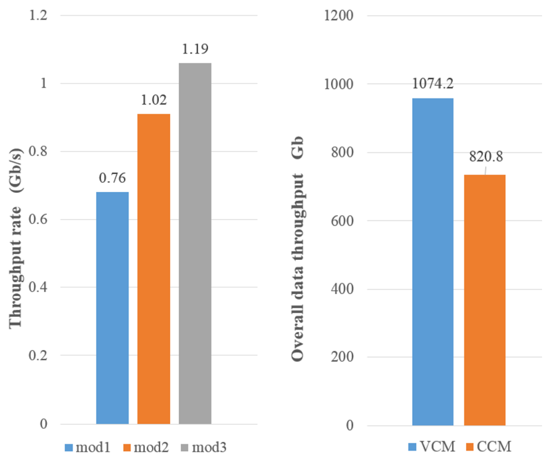

FIGURE 11. **Summary of the FEC encoder core performance.**

As it can be seen from Fig.11, the proposed FEC encoder core can obtain an encoding throughput rate up to 1.19 Gb/s. The overall data throughput of the CCM system adopting a single mod1 mode is 820.8 Gb, while that of the VCM system is 1074.2 Gb, with an improvement of 30.9%.

V. CONCLUSION

This paper proposed an efficient FEC encoder core based on the DVB-S2 standard. Benefitting from the well-designed encoder structure and reuse of computation units and memories, the FEC encoder core is compatible with various VCM modes and lower hardware resources consumed, which represents an excellent compromise between flexibility and hardware utilization. By using parallel architectures of BCH encoder, LDPC encoder, and bit interleaver, low latency and high throughput are achieved. The implementation results show that the FEC encoder core can obtain an encoding throughput rate up to 1.19Gb/s, and the overall data throughput is improved by 30.9% compared with the CCM system, which makes this FEC encoder core a very attractive solution for future VCM-based LEO satellite-ground data transmission systems.

REFERENCES

[1] A. Payandeh, M. Ahmadian, and M. R. Aref, "A secure channel coding scheme for efficient transmission of remote sensing data over the LEO satellite channels," in *Proc. 3rd Int. Conf. Recent Adv. Space Technol.*, Istanbul, Turkey, Jun. 2007, pp. 510–514.

[2] *Variable Coded Modulation Protocol*, document CCSDS 131.5-M-1.4, 2016.

[3] J. Li, W. Xiong, G. Sun, Z. Wang, Y. Huang, and M. Shen, "Doppler-robust high-spectrum-efficiency VCM-OFDM scheme for low Earth orbit satellites broadband data transmission," *IET Commun.*, vol. 12, no. 1, pp. 35–43, Dec. 2018.

[4] D. Capirone, S. Benedetto, G. Montorsi, M. Cossu, R. Roscigno, A. Lupi, M. L'Abbate, C. Riva, and A. Paraboni, "Variable coding and modulation schemes for maximization of low-earth orbit satellite communications," in *Proc. ESA-TTC*, Noordwijk, The Netherlands, Sep. 2010, pp. 1–5.

[5] *Digital Video Broadcasting (DVB); Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, New Gathering and Other Broadband Satellite Applications*, document EN ETSI 302 307 V1.4.1, 2014.

[6] A. Morello and V. Mignone, "DVB-S2: The second generation standard for satellite broad-band services," *Proc. IEEE*, vol. 94, no. 1, pp. 210–227, Jan. 2006.

[7] P. Urard, L. Paumier, V. Heinrich, N. Raina, and N. Chawla, "A 360 mW 105Mb/s DVB-S2 compliant codec based on 64800b LDPC and BCH codes enabling satellite-transmission portable devices," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2008, pp. 310–311.

[8] P. Urard, E. Yeo, L. Paumier, P. Georgelin, T. Michel, V. Lebars, E. Lantreibecq, and B. Gupta, "A 135Mb/s DVB-S2 compliant codec based on 64800b LDPC and BCH codes," in *IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2005, pp. 446–447.

[9] M. Gomes, G. Falcao, V. Silva, V. Ferreira, A. Sengo, L. Silva, N. Marques, and M. Falcao, "Scalable and parallel codec architectures for the DVB-S2 FEC system," in *Proc. APCCAS*, Macao, China, Nov. 2008, pp. 1506–1509.

[10] D. Digdarsini, D. Mishra, S. Mehta, and T. V. S. Ram, "FPGA implementation of FEC encoder with BCH & LDPC codes for DVB S2 system," in *Proc. 6th Conf. SPIN*, Noida, India, Mar. 2019, pp. 78–81.

[11] T. Van Nghia, "Development of the parallel BCH and LDPC encoders architecture for the second generation digital video broadcasting standards with adjustable encoding parameters on FPGA," in *Proc. Conf. EnT*, Moscow, Russia, Nov. 2016, pp. 104–109.

[12] N. Kumar, C. Prakash, S. N. Satashia, V. Kumar, and K. S. Parikh, "Efficient implementation of low density parity check codes for satellite ground terminals," in *Proc. Conf. ICACCI*, New Delhi, India, Sep. 2014, pp. 689–695.

[13] I. Lee, M. Kim, D. Oh, and J. Jung, "High-speed LDPC encoder architecture for digital video broadcasting systems," in *Proc. Conf. ICTCI*, Jeju-do, South Korea, vol. 2013, pp. 606–607.

[14] M. Gomes, G. Falcao, A. Sengo, V. Ferreira, V. Silva, and M. Falcao, "High throughput encoder architecture for DVB-S2 LDPC-IRA codes," in *Proc. Conf. ICM*, Cairo, Egypt, Dec. 2007, pp. 271–274.

[15] X. Liu and Q. Hu, "10Gb/s orthogonally concatenated BCH encoder for fiber communications," in *Proc. 2nd Conf. IMSNA*, Toronto, ON, Canada, Dec. 2013, pp. 1018–1021.

[16] D. Kim, K. R. Narayanan, and J. Ha, "Symmetric block-wise concatenated BCH codes for NAND flash memories," *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4365–4380, Oct. 2018.

[17] G. Hu, J. Sha, and Z. Wang, "High-speed parallel LFSR architectures based on improved state-space transformations," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 3, pp. 1159–1163, Mar. 2017.

[18] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Conf. ISTC*, Brest, France, Sep. 2000, pp. 1–8.

[19] M. Eroz, F.-W. Sun, and L.-N. Lee, "DVB-S2 low density parity check codes with near Shannon limit performance," *Int. J. Satell. Commun. Netw.*, vol. 22, no. 3, pp. 269–279, May 2004.

[20] *CCSDS Protocols Over DVB-S2-Summary of Definition, Implementation, and Performance*, document CCSDS 130.12-G-1, 2016.

JING KANG received the B.S. degree in electronic information science and technology from the Shandong University of Science and Technology, Qingdao, China, in 2016. She is currently pursuing the Ph.D. degree in computer application technology with the University of Chinese Academy of Sciences, Beijing, China. Her primary research interests include channel coding, satellite communications, 5G, and FPGA implementation.

JUN SHE AN received the Ph.D. degree in computer software and theory from Northwestern Polytechnical University, Xi'an, China, in 2004. In 1995, he joined the National Space Science Center, Chinese Academy of Sciences, where he is currently a Professor with the Key Laboratory of Electronics and Information Technology for Space Systems. His current research interests include space integrated electronic technology, space computer hardware and software, system architecture, and data processing technology.

BINGBING WANG received the B.S. degree in communication engineering from Zhengzhou University, Henan, China, in 2018. He is currently pursuing the master's degree in computer application technology with the University of Chinese Academy of Sciences, Beijing, China. His primary research interests include channel coding, satellite communications, and 5G.