Configurable Universal QC-LDPC Encoder Architecture Design

LDPC Encoder

Mingbo Hao[1st], Guangzu Liu[2nd], Jun Zou[3rd], Tian Ban[4th]

School of Electronic and Optical Engineering, Nanjing University of Science and Technology

Abstract

Quasi-cyclic low-density parity-check (QC-LDPC) codes are widely used owning to its good performance and easy hardware implementation. However, most classical LDPC encoder designs do not support changing the LDPC code types or parameters according to different communication services and channel conditions. In this paper, we propose an encoder architecture with configurable code types, input information block lengths, coding rates and degrees of parallelism, which can realize "hot-switching" of configuration parameters. A new scheme that makes full use of the characteristics of the generation matrix is used in the core of parallel coding. The implementation on field programmable gate array (FPGA) shows that the throughput of the encoder can reach 1.6 Gbps under the condition of 7/8 coding rate (8176, 7154) and the degree of parallelism set to 8.

Significance of Configurable LDPC Encoder

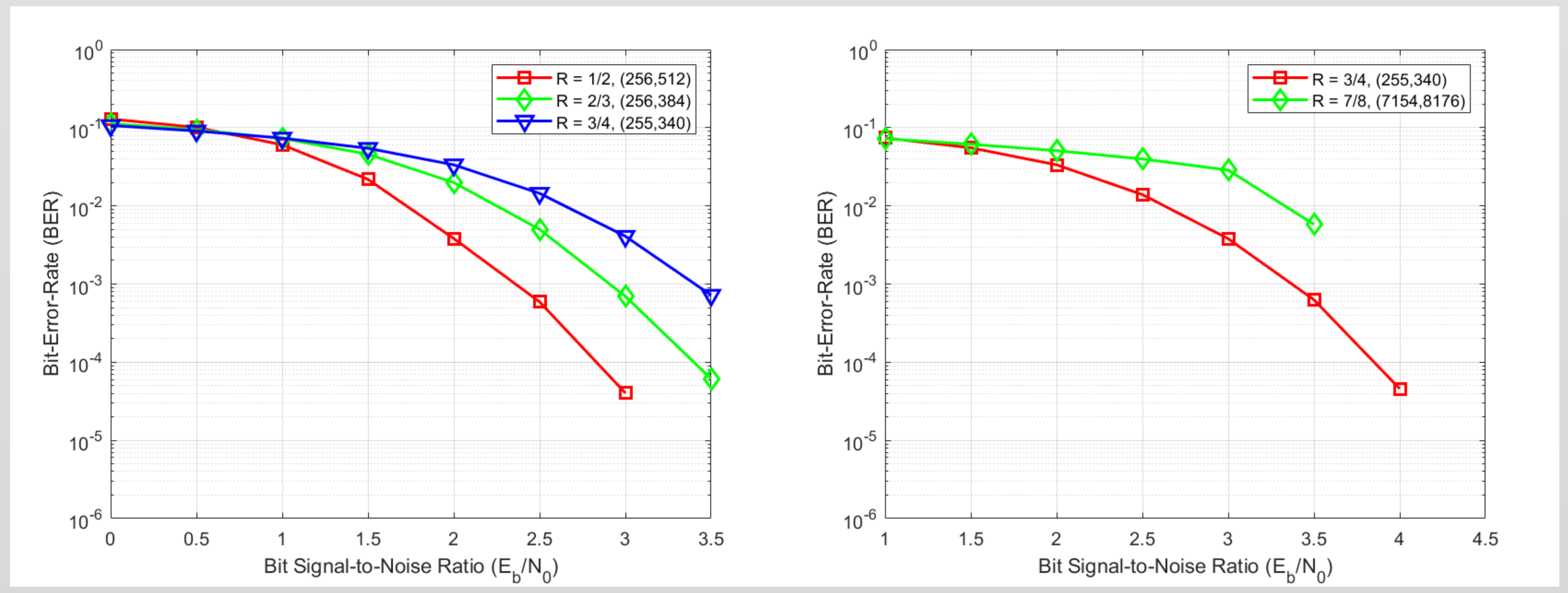

Fig.1 Binary state-shift keying (BPSK) modulation, additive Gaussian white noise channel and normalized min-sum algorithm are used to simulate the performance of information bit length 256 bits with 1/2, 2/3, 255 bits with 3/4 coding rate and 7154 bits with 7/8 coding rate under different signal-to-noise ratio (SNR).

Table Ⅰ. Trade-off on LDPC Codes' Parameter

Perspective	Assessment
Code Type	Can be optimized for different channel conditions…
Coding Rate	Lower coding rate, more check information, but lower data transmission efficiency…
Code Length	Longer code length, better performance, but longer encoding time…
Parallelism	Higher parallelism, higher throughput, but higher resource cost…
Summary	Use configurable LDPC encoder

An Efficient Codeword Computing Unit

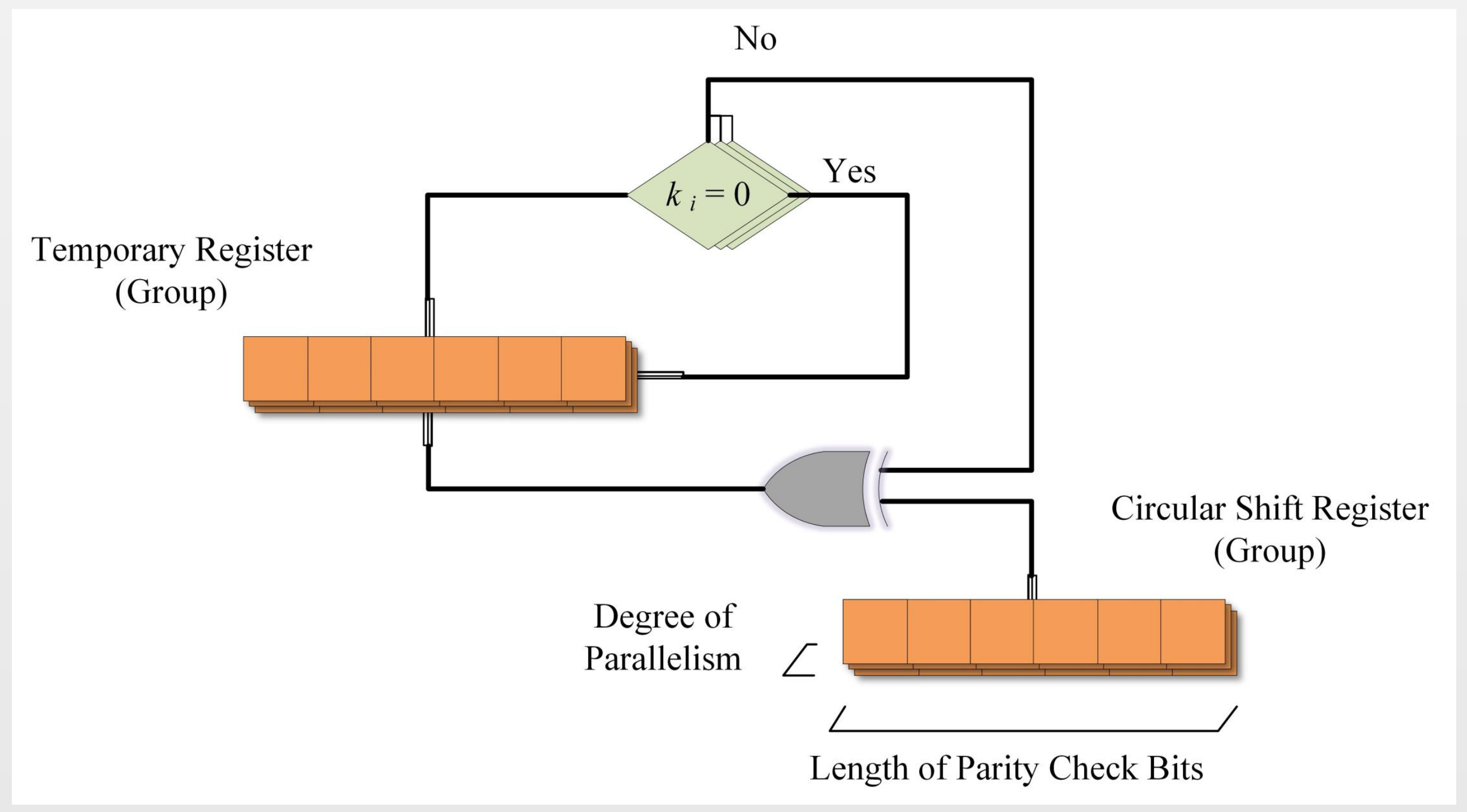
No
Yes
$k_i = 0$
Temporary Register (Group)
Circular Shift Register (Group)
Degree of Parallelism
Length of Parity Check Bits

Fig.2 Efficient coding iteration structure, ki stands for input information bit and in this structure, it can be regarded as a control signal rather than a data stream in the ordinary sense.

- *The width of the register groups depends on the degree of parallelism. For serial coding, of course, only one is required for all components.*

Matrix Crossover in Parallel Coding

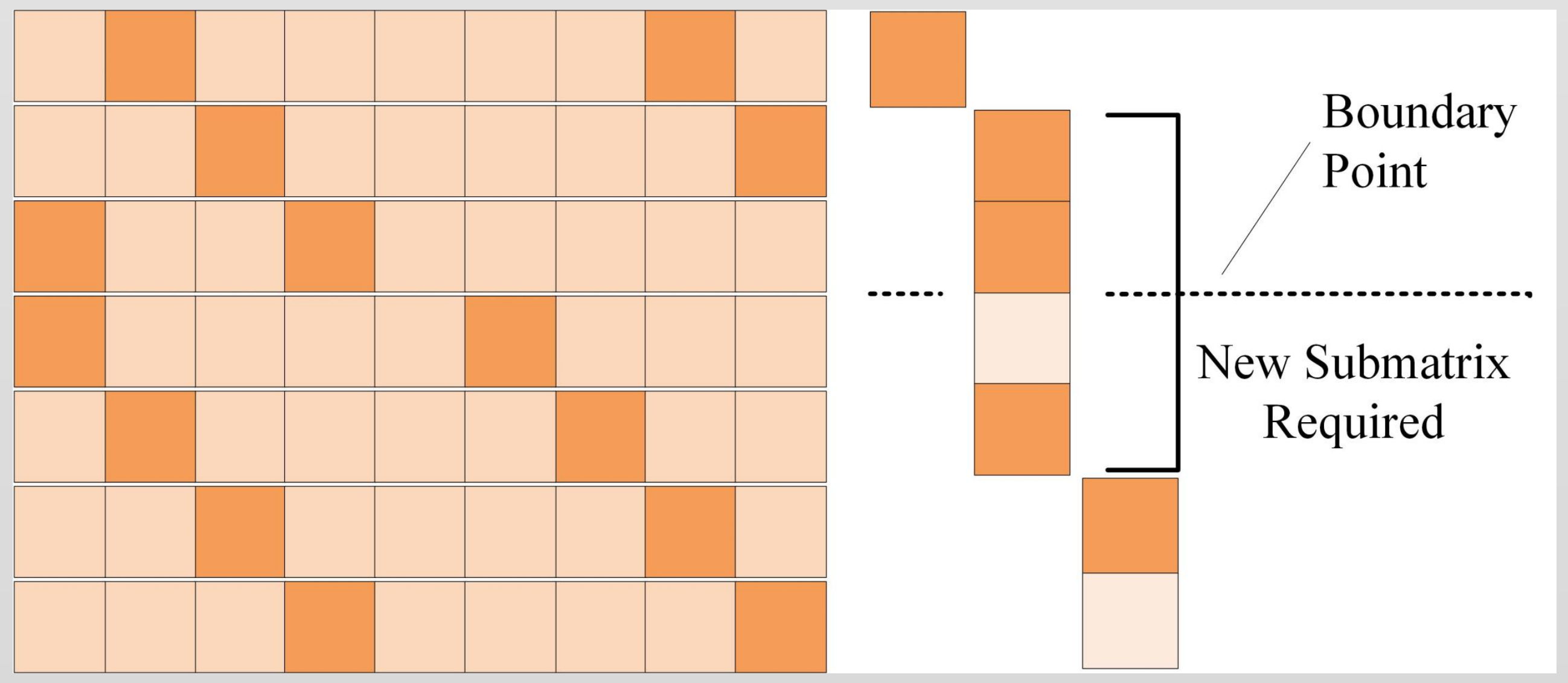

Fig.3 Data crossover phenomenon in parallel coding (Suppose P = 4). The dark color block represents element 1, while the light color represents 0.

- *If the order of the circulant submatrix of the G matrix is not an integer multiple of the degree of parallelism, there will be a crossover phenomenon as described in Fig.3.*
- *To solve this problem, this paper divides the coding section into "boot-area" and "shift-area", and stores the data required by the boot-area in ROM. This results in high clock frequency and high robustness.*

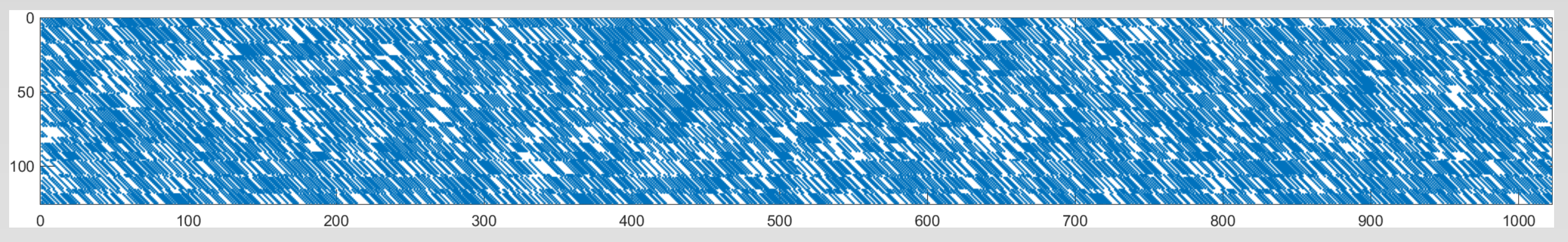

Fig.4 Schematic diagram of boot-area data (Take the LDPC code of 7/8 (8176,7154) coding rate of C2 family as an example, where the parallelism is set to 5).

Coding Control Instruction structure

3 bits	3 bits	2 bits
CT	DOP	CR
7 5	4 2	1 0

Fig.5 Format of control instruction, bit 0 to bit 7 from right to left.

- *Code Type (**CT**)*
- *Degree Of Parallelism (**DOP**)*
- *Coding Rate (**CR**)*

Encoder architecture

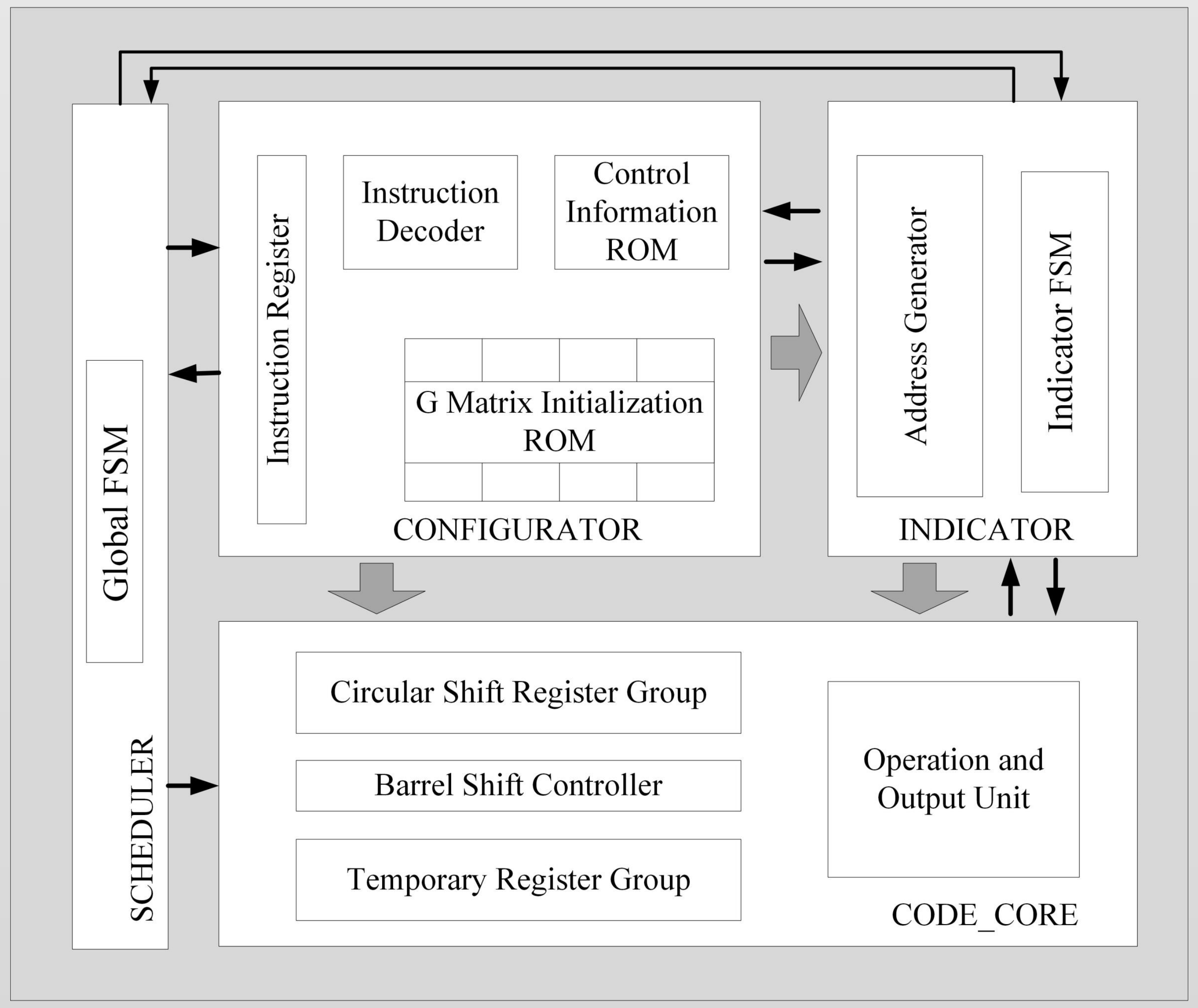

Fig.6 Schematic diagram of encoder architecture, where the thick arrow represents data stream and the thin arrow represents control signal.

- *In order to realize dynamically configurable LDPC coding, there are four states defined in SCHEDULER's finite state machine, which are: idle (**IDLE**), instruction fetch (**IF**), configure (**CONFIG**) and encode (**ENCODE**), and the operation of the module is scheduled according to different states (e.g., Instruction fetching can only be carried out in the IF state). After completing one encoding, the system will return to the IDLE or IF state according to the coding enable signal EN_CODE. When the state of the system jumps from CONFIG to ENCODE, it must ensure that the valid signal of configuration VALID_ CONFIG had asserted.*