FPGA-Based Large Constraint Length Convolution Code Encoder Verification

Zhang Taotao¹, Zhang JingKe², Zhou Zhiwen³, Yang Zhifei¹, Liu Wanhong¹
¹China Luoyang Electronic Equipment Test Center
²State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System
³Command College of PAP

Abstract. The convolution code with large constraint length plays an irreplaceable role in deep space communication and ultra-low frequency communication. Therefore, it is very important to find and test convolution code with large constraint length. Convolutional codes are widely used in deep space communication systems because of their high coding gain and simple and reliable encoders. The performance and implementation difficulty of convolutional codes mainly depend on the constraint length of the decimal codes and the coding efficiency. Enlarging and improving the coding gain of convolutional codes will greatly increase the complexity of decoders. The method of mathematical derivation and verification by mathematical tools is not suitable for convolutional codes with large large constraint length. In order to improve the efficiency of inspection, the method of parallel multi-core computing needs to be introduced into the evil code test. Tests show that the FPGA-based parallel inspection method can improve the test efficiency by geometric multiples.

1. INTRODUCTION
Convolutional codes were first proposed by Elias in 1955. Then Wozencraft and Reiffen proposed a sequence decoding method, which is very effective for convolutional codes with about beam length. In 1963, Massey proposed threshold decoding, which made convolutional codes widely used in digital transmission of satellite and wireless channels. In 1967, Viterbi proposed a maximum likelihood decoding algorithm, which is easy to implement soft decision decoding of convolutional codes with small constraint length. Viterbi algorithm combined with soft decision of sequence decoding makes convolutional codes widely used in deep space and satellite communication systems in the 20th century. [1]

Because convolutional codes make full use of the correlation between the code groups, it has been proved that the performance of convolutional codes is not inferior to or even better than block codes in theory and practice. However, unlike block codes with strict algebraic structure, convolutional codes have not yet found a rigorous mathematical means to regularly link error correction performance with code composition. At present, most of them use computers to search for good codes. At the same time, the search code must be checked to eliminate the interference of malignant convolutional coder. Research on search and verification of large constrained length convolutional codes is still in progress. [2,3].
2. EVIL ENCODER

Fig 1. Convolutional code encoder
There are many ways to describe convolutional codes. Generally speaking, they can be divided into two algebraic and graphical methods. The algebraic method and state transition diagram are used to describe (2, 1, 3) coders. The general structure of convolutional coder is shown in Figure 1. A frame of parallel data formed by serial and parallel converters is sent to the linear logic unit, and the data is sent to the level data frame shift register. The encoding logic obtains the encoding output by linear logic operation based on the current input data and the previous data stored in the data register, and then converts it into serial output through parallel and serial transformation.

The generating polynomial of the encoder shown in Figure 1 is given by (1).

\[ G(D) = \frac{g_0(D)}{g_1(D)} = [1 + D + 1 + D^2] \]  

(1)

Maximum common factor for generating polynomials

\[ GCD[g_0(D), g_1(D)] = GCD[1 + D + 1 + D^2] \]  

(2)

The structure of the encoder determines that even if the input information sequence has an infinite code weight, the output information sequence of the encoder has a fixed code weight of 3.

Fig 2. Convolutional code state diagram

The state of the encoder is defined as the storage value of the register. The current state and output are uniquely determined by the previous state and the current input. When a set of input information is moved into the encoder, the state of the encoder is transferred. The memory value of the register at any time in the convolutional encoder shown in Figure 1 (2, 1, 3) is referred to as a state of the encoder, which is represented by. Figure 2 shows the input of the encoder in the state transition with the upper digit on the diagonal line and the output of the state transition with the lower digit. As can be seen from Figure 2, if the initial state of the encoder is , the encoder will be transferred from state to state when input 1 information element, and output subgroup 11. If input 0 information element, it still stays in state, output subgroup 00, and so on. With the continuous input of information elements, the state of the encoder is constantly shifting, and the corresponding branch subgroups are output, thus forming a code sequence corresponding to the input information sequence. Notice that when the input information element of state is 1, the output is 00. This is a malignant encoder. When the code weight of the input information element is infinite, there is a zero weight ring. [4].

3. PARALLEL COMPUTATION METHOD FOR FPGA TEST

Considering that the evil convolutional coder has an important characteristic that there is at least one zero weight ring around the zero weight ring around the initial state \( S_0 \). With the help of the computer traversing all the states of the convolutional encoder, we can get a convolution code polynomial
checking algorithm based on this characteristic.

The single state checking algorithm of convolutional encoder \((n,1,v)\) is as follows:

1. The status of the current register is randomly selected to be placed in a shift register with \(v\) storage cells;
2. Every time one bit is shifted to the left, it is judged whether it has returned to the state at the beginning, the highest bit of the register is shifted out to the last bit of the register, calculate the output code weight, the output code weight is calculated, and it is determined whether to perform the next operation according to the output code weight;
3. If the weight of the output code word is always 0, the operation in (2) is repeated until \(v\) times, and the state in the register is returned to the state at the beginning after \(v\) cyclic shift. The shift operation ends and the cycle is exited;
4. When the shift operation is finished, if the Hamming weight is 0, then the encoder is a evil encoder, otherwise the state can not prove that the encoder is malignant.

The full state checking algorithm of convolutional encoder is as follows:

1. Starting from the initial state;
2. Check each state separately;
3. If the result of the test can indicate that the encoder is malignant, jump out of the loop, otherwise let the state add one to the next state;
4. The operations in (2) and (3) are repeated until the loop is exited when the binary representation of the state is all one (end state), and the test algorithm ends.

![Fig. 3. The test algorithm flow chart of convolutional code encoder](image)

The convolution encoder verification algorithm shown in Fig. 3 is a software implementation scheme[5], so it can not be directly applied to hardware implementation. With the increase of constraint length, the computation complexity of encoder test is increasing. To test a convolutional code of a larger constraint length, more states must be checked, and full hardware engineering techniques can be used. In order to apply the verification algorithm to hardware implementation, the state machine must be designed to classify and merge the flowcharts. Then the hardware structure of the encoder is designed, and the logic design of the convolutional encoder checker is obtained. Therefore, the flow chart of the convolutional encoder verification algorithm is modified to adapt to the hardware structure of the scheme. The flow chart of the improved convolutional code encoder checking algorithm is shown in Figure 4[6].
**4. STATE MACHINE DESIGN**

According to the flow chart of the convolutional code encoder shown in Fig. 4, the conversion relationship between the states is shown in Fig. 5.

The state transition diagram indicates any state in the state to be detected, $S_{n+1}$ represents the next state of state $S_n$, $i/x$ means that the output is $x$ when $i$ is input. Suppose the output codeword of the 1/2 encoder is $q_0$ and $q_1$, $x = q_0 \& q_1$. According to the logical relationship between $x$ and $q_0$ and $q_1$, the corresponding values can be obtained, $\&$ means and. Each work unit processes a piece of state, the state $S$ in the register needs to be circularly shifted $\nu$ times. The output code weight of the coded polynomial is determined after each cycle shift, as long as the code weight is not zero after any of the cycles of $\nu$, it is shown that the coding polynomial is non-malignant in this state, the state of the register in the processing unit is transferred to the next state, until the end of the state processed by the unit.

The timing diagram clearly describes the transition relationship between states of the state machine under different conditions, when the detected encoder is non-malignant, the state machine's working state is constantly switching between the initial state and the cyclic calculation, it will not end with the end of inspection, when the detection task is completed, the signal will be given by the display lamp, the task of a state machine is to check the status of registers; When the coder being tested is malignant,
the detected state can prove that the encoder is malignant, the state in the state machine is no longer switched but remains the current state, gives a signal that confirms that the current encoder is malignant. Figure 6 shows the time sequence diagram of the state machine working in different states.

Figure 6 shows the time sequence diagram of the state machine working in different states.

Sequence of diagram of state machine under non-malignant encoder

- Set initial state
- Cyclic calculation
- Set initial state
- Cyclic calculation

Sequence of diagram of state machine under malignant encoder

- Set initial state
- Cyclic calculation

Fig. 6. The state machine work sequence diagram

When each processing unit in the FPGA starts working, it needs to be initialized first. After the initialization work is completed, the calculation is made according to the set conditions. Figure 7 shows the schematic diagram of the transformation of the state machine.

When the input control signal is 1, it takes one clock cycle to enter the next state to be detected, hen the input control signal is 0, it takes one clock cycle to shift the current state to the right by one bit.

Fig. 7. The state machine transition diagram

REFERENCES

[1] Ranjan Bose. Coding and Cryptography[M]. NewYork: McGraw-Hill Companies. 2002: 60-80.
[2] Vander Poel A P M. A Computer Search for Good Convolutional Codes[J]. IEEE Transaction on Information Theory. 1971, 19 (10): 625-638.
[3] Kennedy J, Eberhart R. Particle Swarm Optimization[M]. New York: Springer US, 2010: 770-776.
[4] Lin Shu, Costello D. Error Control Code (Second edition)[M]. Beijing: Machinery Industry Press, 2007.
[5] Zhang Taotao, Jiang Yuzhong, Liang Zhen. Malicious Code Inspection Algorithm for Parallel-Computing Convolution Code [J]. Communication technology, 2013, 46(9): 26-28.
[6] Zhang Taotao, Wanag Manxi, Rong Hui. The FPGA Test Algorithm Of Large Constraint Length Convolution Code Encoder[C]. ChongQing:IAEAC 2018,2018:2180-2184.