A configurable BCH encoder and decoder

Merlin Ji

The doc describes an IPCore in verilog about binary BCH encoder and decoder. BCH is a popular error correcting code used in storage and transmission system. It adds some redundancy check data into original data frame, the redundancy data length depends on correcting capacity, and all the calculation proceed in the Galois Field that is suitable for FPGA.

1. Algorithm of BCH

There are three steps in design.

1.1 First is selecting eigenpolynomial and the order of generated polynomial. The formula is listed as below.

L1+L2<=2^N (1)

L2 = N*T (2)

L1 is the bit length of the original data. L2 is the bit length of the redundancy check data. N is the order of the eigenpolynomial. T is the correctable bit threshold.

L1 and T depend on system requirement. According to L1 and T, get L2 and N.

1.2 Second step is encoding. According to eigenpolynomial, the Galois Field is generated, and T-1 generated polynomials are calculated. Every generated polynomial takes 2*T-1 power of the primitive element as the root. Then encoding polynomial is obtained by multiplication of all the generated polynomial and eigenpolynomial. Finally, redundancy check bits are generated. The calculation is a polynomial division. The original data is divided by the encoding polynomial, and the remainder is the redundancy check bits. The data frame is re-format by attaching original and redundancy data.

1.3 Last step is decoding. The decode process is more complicated than encode. There are three steps in decode.

1.3.1 syndrome calculation

There are 2*T-1 syndromes required. Firstly get odd order ones. The calculation are T times' polynomial divisions. There are T polynomials that are T-1 generated polynomials and eigenpolynomial. The received data is divided respectively by T polynomials, and remainders are syndromes corresponding to 2*t-1 power of primitive element, t is from 1 to T. Secondly, even order syndromes are calculated by odd ones. According to the relationship between t power of primitive element and 2*t power of primitive element in Galois Field, 2*t syndrome is transformed by t syndrome. Finally, the 2*T-1 syndromes are all obtained.

1.3.2 error position polynomial calculation

The error bit position polynomial is calculated by inversionless BM algorithm.

The calculation is iterated at most 2*T-1 times. Define S as syndrome whose order is 2*T-1. Define v is the error position polynomial whose order is also 2*T-1. There are some variables in calculation. First of all, initialize all the variables.

$v^{(0)}=1$，$k^{(0)}=1$，$\delta^{(-2)}=1$

define k as iteration times，start counting from 0, end at T times.

Define $d^{(2k)}$ as the coefficient of $z^{(2k+1)}$ in the product $(1+S)v^{(2k)}$, the iteration proceeds as 3 formulas below.

$$v^{(2k+2)} = \delta^{(2k-2)} v^{(2k)} + d^{(2k)} k^{(2k)} z \tag{3}$$

$$k^{(2k+2)} = \begin{cases} z^2 k^{(2k)}, & \text{if } d^{(2k)} = 0 \text{ or if } \deg v^{(2k)} > k \\ z v^{(2k)}, & \text{if } d^{(2k)} \neq 0 \text{ and if } \deg v^{(2k)} \le k \end{cases} \tag{4}$$

$$\delta^{(2k)} = \begin{cases} \delta^{(2k-2)}, & \text{if } d^{(2k)} = 0 \text{ or if } \deg v^{(2k)} > k \\ d^{(2k)}, & \text{if } d^{(2k)} \neq 0 \text{ and if } \deg v^{(2k)} \le k \end{cases} \tag{5}$$

1.3.3 error position search

The chian search method is taken to find error positions. Every element is put in Galois Field into the error position polynomial, if the result is equal to zero, the element corresponds with error position. The search could proceed parallel in order to shorten run time.

2. Design of IPCore

The module of the IPCore is depicted as below. The original data generate redundancy check data according to the correcting requirement, and then a new data frame including original and check data transmits. Some bits could be reversed in the channel. The frame of data generate syndromes in the receiver. And then getting polynomial of error bit position by bm algorithm. Next process is substituting every element into the polynomial to find error position. The process could be parallelized in order to speed up searching. Last step is correcting error bit and correcting information report.

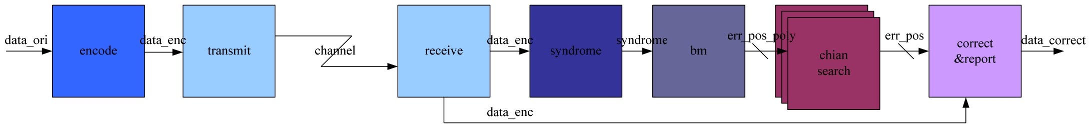

Figure 1 module of bch encoder and decoder

3. feature of IPCore

The IPCore is configurable with some parameters. The parameters are input data width, eigenpolynomial and its order, correcting capacity, data frame length, search parallelism. The parameters are basic system information, easy to set, and avoid complicated calculation.

4. performance

TBD