CRC Algorithm and its General Encode Development

Hou zhi
Changchun University of Science and Technology
Changchun, China
houzhi1990@sina.com

Yang Xiaohui*
Changchun University of Science and Technology
Changchun, China
yangxiaohui1963@sina.com
* Corresponding Author

Abstract— The traditional CRC encoder is usually only for a kind of stationary generating polynomial, so the signal sending end and the receiving end is the one to one relationship. To break this relationship, we want to add some control to make generating polynomial can be artificially easily changed. Finally, achieve the circuit design by FPGA and simulate by Modelsim. This paper mainly designed the principle general of CRC encode which generating polynomial can be changed, derive CRC model for all polynomial and simulate the results.

Keywords-CRC; Polynomial; General CRC; FPGA; Modelsim

I. INTRODUCTION

Digital communication process often affected by noise and other complex factors. It makes the data changed after channel. The changed data are error codes, check code plays an important role in error control.

In order to catch these error codes, the sender perform a mathematical operation on the information data which will be sent. Send out the data together with the result of the operation [1] [2]. Then perform the same operation on the receive data and compare the result. If the information data are corrupted during transmission, the two result will inconsistent. The receiver will apply for the sender send data again. This is error control [3] [4].

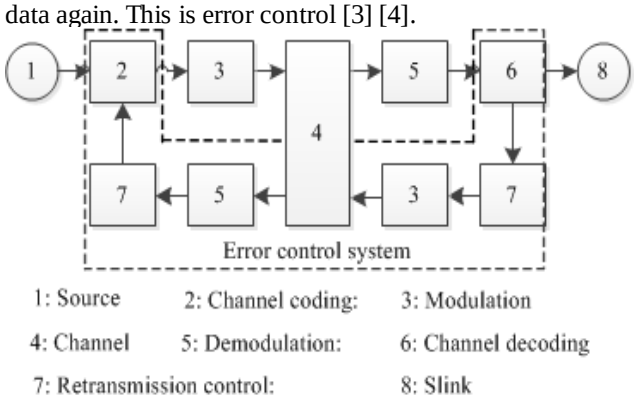


Figure 1. Principle of Error Control

II. PRINCIPLES AND MODEL

A. The Principle of CRC

In short, CRC check principle is followed a K bits information data sequence with an R bits check code (sequence). Thereby build an N (N= K + R) bits binary sequence. There is a unique relation between the check code and information data. If one bit or some bits of the data are changed, this particular relationship will be destroyed. Therefore, we can judge the data changed or not [5].

B. Algorithm of CRC

The binary sequence which will be send is shown as follow:

$$V(x) = A(x)g(x) = X^R M(x) + r(x) \quad (1)$$

where $M(x)$ = information data polynomial (degree K); $r(x)$ = check code polynomial (degree R); The send binary sequence form of data sequence and check code sequence.

$$M(x) = M_n \cdot 2^n + M_{n-1} \cdot 2^{n-1} + \cdots + M_1 \cdot 2^1 + M_0 \quad (2)$$

As shown in above, $r(x)$ can be obtained by calculation. First, the information data polynomial is multiplied by 2^R (binary sequence fill R "0"), then divided by the generator polynomial $g(x)$ [6].

$$\frac{M(x)\cdot 2^R}{g(x)} = \frac{M_n \cdot 2^R}{g(x)} \cdot 2^n + \frac{M_{n-1}\cdot 2^R}{g(x)} \cdot 2^{n-1} + \cdots + \frac{M_1 \cdot 2^R}{g(x)} \cdot 2^1 + \frac{M_0 \cdot 2^R}{g(x)} \quad (3)$$

Make:

$$\frac{M_n \cdot 2^R}{g(x)} \cdot 2^n = Q_n(x) + \frac{R_n(x)}{g(x)} \quad (4)$$

$Q_n(x)$ is an integer, representing the quotient, $R_n(x)$ is a binary remainder. Put the (3) into the (2) as follows:

$$\begin{aligned}\frac{M(x)\cdot 2^R}{g(x)} &= \left\{Q_n(x) + \frac{R_n(x)}{g(x)}\right\} \cdot 2^n + \frac{M_{n-1}\cdot 2^R}{g(x)} \cdot 2^{n-1} + \cdots \\ &\quad + \frac{M_1 \cdot 2^R}{g(x)} \cdot 2^1 + \frac{M_0 \cdot 2^R}{g(x)} \\ &= Q_n(x) \cdot 2^n + \left\{\frac{R_n(x)\cdot 2}{g(x)} + \frac{M_{n-1}\cdot 2^R}{g(x)}\right\} \cdot 2^{n-1} + \cdots \\ &\quad + \frac{M_1 \cdot 2^R}{g(x)} \cdot 2^1 + \frac{M_0 \cdot 2^R}{g(x)} \end{aligned} \quad (5)$$

Use $Q_i(x)$ replace $M_i(x)$ until the end:

$$\frac{M(x)\cdot 2^R}{g(x)} = Q_n(x) \cdot 2^n + Q_{n-1}(x) \cdot 2^{n-1} + \cdots$$

$$+Q_0(x)\cdot 2+\frac{R_0(x)}{g(x)} \quad (6)$$

$$r(x)=R_0(x) \quad (7)$$

The $R_0(x)$ is check code we need [7].

C. Modulo Two Division

All division above are modulo two division. The difference of modulo two division and ordinary division is remainder. The remainder of modulo two division is equal to *XOR* value of the dividend and divisor [8]. The ordinary is equal to the difference.

Allow information to be send M = 1101110011, generating polynomial is x^5+x^3+1. The corresponding code g = 101001, R = 5. After M fill five 0,Then do the modulo two division for g, Remainder $r(x)$ = 01100. The calculation process is shown in figure 2. Therefore, the actual data to be sent is 110111001101100.

D. The Structure of CRC Encoder

How to get the check code from information data is the key of the encode. The structure of the encode including division circuit and other auxiliary circuit. Division circuit was composed of shift register and modulo two adder [9].

$$\begin{array}{r} 1110001100 \\ g \rightarrow 101001\overline{)1101110011\ 00000} \\ 101001 \\ \hline 111100 \\ 101001 \\ \hline 101010 \\ 101001 \\ \hline 111100 \\ 101001 \\ \hline 101010 \\ 101001 \\ \hline 01100 \leftarrow r(x) \end{array}$$

Figure 2. Modulo Two Division Arithmetic Calculation Process

In CRC5 example, the structure of CRC5 is shown as follow:

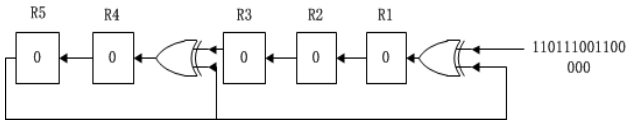
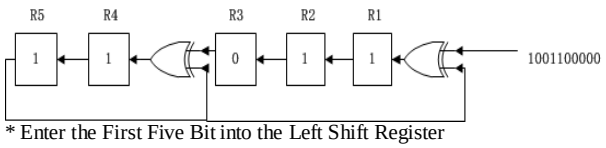


Figure 3. The 0 Step of Enter Data



* Enter the First Five Bit into the Left Shift Register

Figure 4. Step 1 to Step 5 of Enter Data

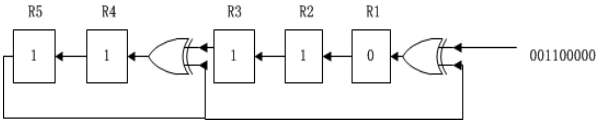


Figure 5. The Sixth Step of Enter Data

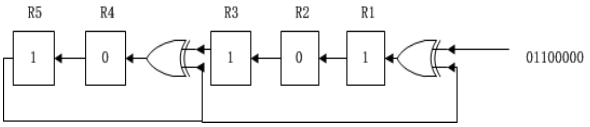
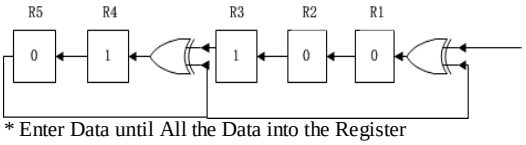


Figure 6. The Seventh Step of Enter Data



* Enter Data until All the Data into the Register

Figure 7. The Last Step of Enter Data

The encode needs to meet the follow requirements [10]:

- The number of the shift register in division circuit is equal to highest degree(R) of $g(x)$;
- If a bit of the coefficient of $g(x)$ is equal to 0, the corresponding register XOR gate is invalid.
- The number of bit need fill "0" is equal to R (degree of generator polynomial).

III. GENERAL CRC

A. Design of General CRC Rncoder

The Species of generator polynomial is as many as forty or fifty, Table. 1 list some of them.

TABLE I. EXAMPLES OF COMMONLY USED GENERATOR POLYNOMIAL

Name	Polynomial	Use
CRC-5-USB	x^5+x^2+1	USB signaling packet
CRC-8-ATM	x^8+x^2+x+1	ATM; HEC
CRC-12	$x^{12}+x^{11}+x^3+x^2+x+1$	Communication system
CRC-16-IBM	$x^{16}+x^{15}+x^2+1$	Bisync; Modbus; Usb
CRC-16-DECT	$x^{16}+x^{10}+x^8+x^7+x^3+1$	Cordless telephone
CRC-24	$x^{24}+x^{22}+x^{20}+x^{19}+x^{18}+x^{16}+x^{14}+x^{13}+x^{11}+x^{10}+x^8+x^7+x^6+x^3+x+1$	Flex Ray
CRC-30	$x^{30}+x^{29}+x^{21}+x^{20}+x^{15}+x^{13}+x^{11}+x^8+x^7+x^6+x^2+x+1$	CDMA
CRC-64-ISO	$x^{64}+x^4+x^3+1$	HDLC—ISO 3309
CRC-128	Replace*	
CRC-160	Replace*	

* IEEE-ITU Standard. Replace by MD5 & SHA-1.

The survey found that although the degree of generator polynomial can be as high as 128 or even 160, the largest degree of we can use is 64. CRC encoder needs to have the following characteristics to meet the requirements of the encoder to achieve the goal of general.

- A 64 bits shift register. If the chosen generator polynomial degree is largest, the number of shift register is still enough.
- All registers are connected with an XOR gate. It is effectively controlled by AND gate.
- No matter how many degree of the generator polynomial is, the data all is filled by sixty-four "0". The ultimate goal is to make the last bit of the data also can pass the shift register.

B. *The Structure of General CRC Encode*

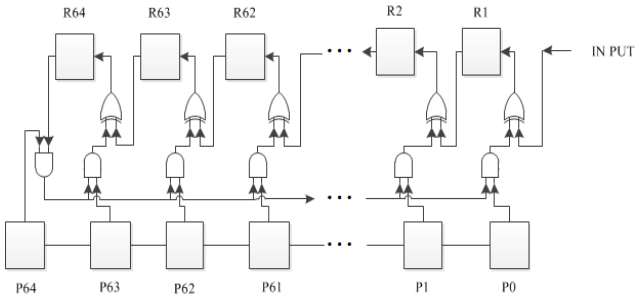


Figure 8. The Structure of General CRC

The R1~R64 are 64 bits shift register, P0~P64 are 65 bits register is used to hold generator polynomial. Let R64 and all polynomial register do logic AND arithmetic. Then, the results and shift register do XOR operation. Finally, sent the results of the XOR operation into the next bit of the shift register.

C. *Work Flow of General CRC Encoder*

At the beginning, input generator polynomial into generator polynomial register, and calculate the number of bit of the polynomial (N) (It is aim to calculate how many bits of the check code is (R = N-1). We want to choose the highest R bits of register as a check sequence).

After the generator polynomial input finished, began to input information data into shift register. When the information data input finished, it need entering sixty-four 0 to register, then highest R bits of register is what we need.

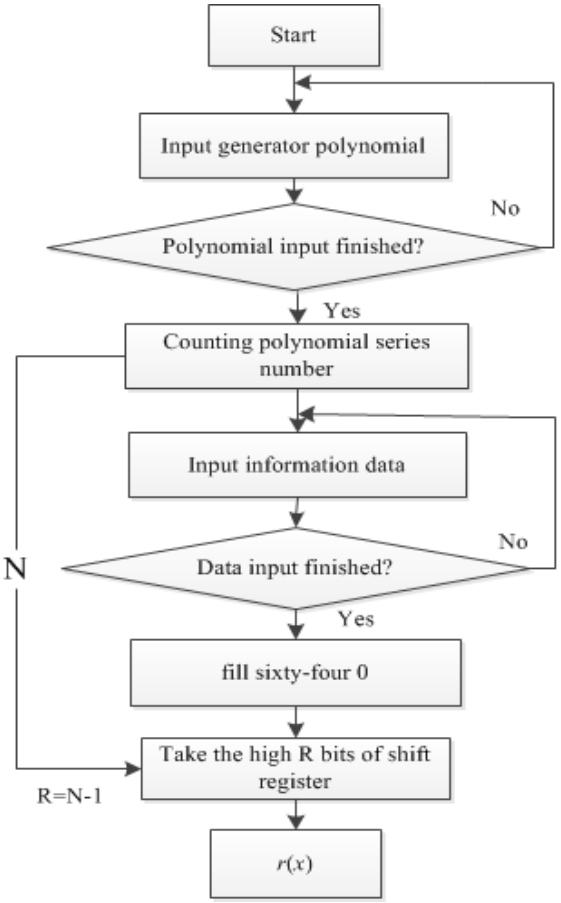


Figure 9. Flow Chart General CRC

IV. SIMULATION

Let generator polynomial is 1010 01ZZ (The input port is an 8 bit parallel input, if the bits of input are less than eight, the low bits should fill with "Z" . So 1010 01ZZ is equal to 10 1001). The information data is 1101 1100 11ZZ ZZZZ (11 0111 0011). Result is 0110 0ZZZ (0 1100). It is same to the result of the analysis. Waveform simulation as follow:

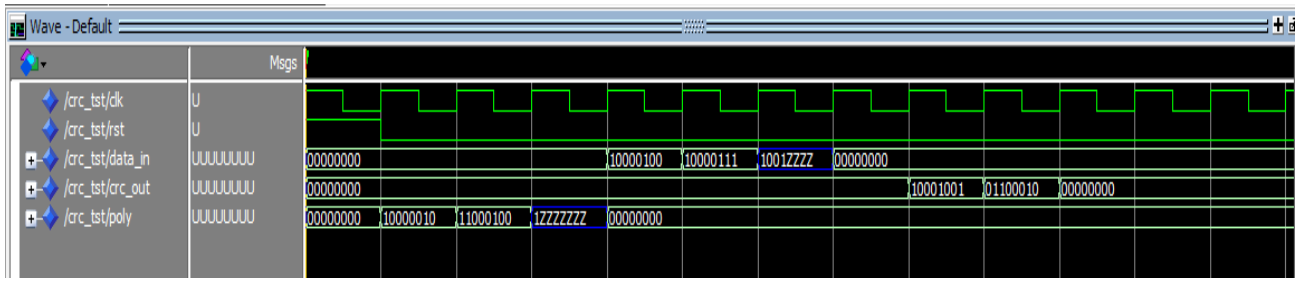


Figure 10. Simulation 1

In another generator polynomial (1 0000 0101 1000 1001) simulation:

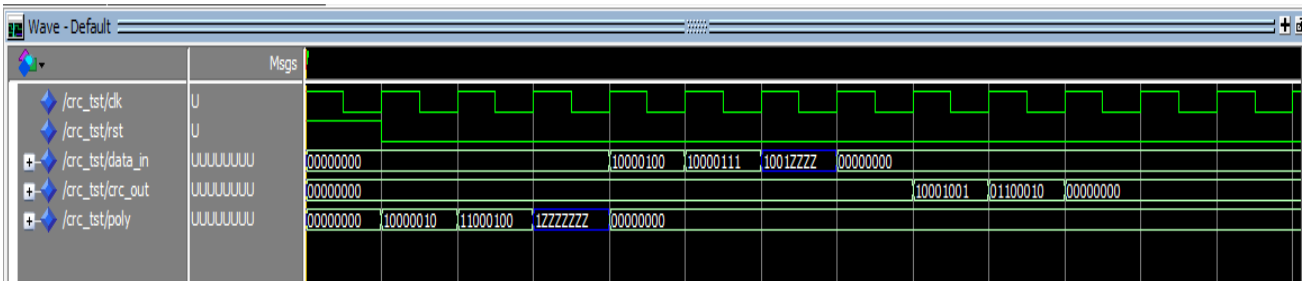


Figure 11. Simulation 2

V. CONCLUSION

General CRC has many advantages. Such as, the structure is simple, the algorithm is easy to understand. It can be applied to most of the generator polynomial, so it meet the different needs of different areas. But it also has many shortcomings. It caused by the disadvantages of CRC algorithm. For example, the different information data may be have the same check code. So, it cannot detect information data transmission errors effectively.

VI. REFERENCES

[1] Campobello G and Patane G and Russo M. "PARALLEL CRC REAL-IZATION". IEEE Transactions on Computers. 2013.

[2] Huang X. "Realization of cyclic redundancy check (CRC) based on Verilog" [J]. Journal of Jinan University. 2010(01).

[3] Su. LH. "Information theory and error correction encoding" [M], Electronics Industry Press.2011.

[4] Peter Sweeney and Yu Y and Zhang DZ. "Error control encoding" [M], Tsinghua University press.2009.

[5] Tao WG and Jia ZN. "Iterative detection method for modulation system with error control of encoding differential unitary space" [J]. Television Technology .K. Elissa, "Title of paper if known," unpublished. 2013(23).

[6] Wang YQ and Yang HX. "Research and implementation of CRC code string" [J]. Computer technology and development.2014 (06).

[7] Zhang ZP and Cheng ZL and Xiao L. "Built in parallel CRC verification of FPGA based on UART" [J]. Automation and instrumentation.2013 (03).

[8] Miao YC and Shen BS and Dou JJ. "Modern communication principle and Application" [M], Electronics Industry Press, 2012.

[9] Williams Ross and a Painless. "Guide to CRC Error Detection Algorithms" [OL]. http://www.repairfaq.org/filipg/ 2010

[10] SHUKLA S and BERGMANNNW. Bergmann. "Single bit error correction implementation in CRC-16 on FPGA". IEEE International Conference on Field Programmable Technology, 2014.

