The research of parallel CRC pipeline algorithm based on matrix transformation

Xiaohui Wang,Liyun Zhuang,Qing Lu,Shihu Wang
Faculty of Electronic and Electrical Engineering
Huaiyin Institute of Technology
Jiangsu Huaian, China
e-mail:allenwxh@163.com

Abstract—**According the universal serial cyclic redundancy check (CRC) technology, one of the new CRC algorithm based on matrix is refered,which descibe an new parallel CRC coding circuit structure with r matrix transformation and pipeline technology. According to the method of parallel CRC coding in high-speed data transmiting, it requires a lot of artificial calculation. Due to the large amount of calculation, it is easy to produce some calculation error. According to the traditional thought of the serial CRC, the algorithm of parallel CRC based on the thought of matrix transformation and iterative has been deduced and expressed. The improved algorithm by pipeline technology has been applied in other systems which require high timing requirements of problem; The desin has been inplemented through Verilog hardware description language in FPGA device, which has achieved a good validation. It has becomed a very good method for high-speed CRC coding and decoding.**

Keywords-CRC;pipeline;FPGA;matrix

I. INTRODUCTION

Cyclic Redundancy Check (Cyclic Redundancy Check, CRC) technology can be used to test the integrity of the process in data transmission or data compression. It has been widely used in the communication network and the data storage technology, etc. According to relevant communication protocol, CRC check code usually locates in the beginning or ending of the frame. The receiver calculates the data of received from transmiter with the same CRC algorithm. If the value calculated equals to checking code in frame, it shows that the process of transmission or compression is successful. Classic CRC algorithm is realized through the structure of linear feedback registers. Because It is a serial coding method, it is intuitive and easy to realize.But the speed of calculate is very low,according to the classic CRC algorithm.It is not suitable to apply in high speed data transmission. Parallel CRC algorithm can meet the demand of the high-speed data transmission.The general method of the parallel CRC is realized by software or hardware. Because speed of the software calculation is slow, the algorithm has been used in many communication equipment by hardware for hign-speed. According to the thoughts of traditional serial CRC algorithm, parallel CRC PIPELINE ALGORITHM based on MATRIX is refered for high-speed coding. The algorithm can reduce the delay time of circuit and improve the throughput rate of data.

II. A SERIAL CRC ALGORITHM

As shown in figure 1, the typical implementation methods of serial CRC algorithm is realized by LSFR, which include shift, feedback, mode 2 division and some multiplication and so on. The Model 2 division is only effective, when the divisor is binary number "1" in the process of mould 2 division. So it is similar to a kind of XOR operation in digital circuits, For binary data flow, it can use mode 2 polynomial to said. For example, 110011 can be expressed as $1\times x^5+1\times x^4+0\times x^3+0\times x^2+1\times x^1+1$, which can be simplified as x^5+x^4+x+1 . Assuming that the data stream U(x), CRC polynomial G(x),quotient Q(x) and remainder R(x). Modulo 2 division can be expressed as: U(x)/G(x)=Q(x)+R(x)/G(x).The specific algorithm is described as following: the order of G(x) is r, the first U(x) multiplied by x^r and then divided by G(x) and the quotient is the R(x). The purpose of multiplied by x^r is to ensure that the order of final quotient is r- 1. The structure of hardware implementation is illustrated in Figure 1:

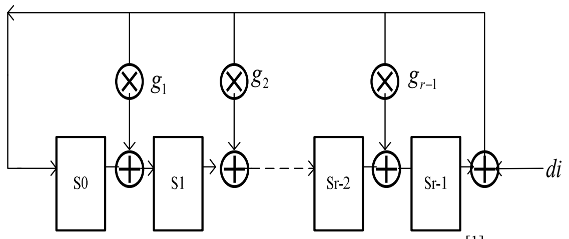

Figure 1 CRC coding circuit structure[1]

III. THE ALGORITHM OF THE PARALLEL CRC WITH MATRIX

According to the hardware structure Parallel CRC coding algorithm in Figure 1, the input data take the mode 2 division and the right shift bit by bit. After m clock, all of the m data have been calculated. The output value of each register will compose parallel m-bit checksum. Assuming that the order of CRC polynomial G(x) is n and the bite width of input binary data U is m bits, the polynomial can be expressed as G(x)= $\sum_{i=0}^{i\le n-1} g_i(x)x^i$ .The output value of each D flip-flop is the result of division and shift between CRC polynomial and the input binary data in a certain period of time, which is taken as the result of CRC calculation. Assuming that $crc[n-1]_i$ expresses the out value of $crc[n-1]$ at time I and the d_i expresses serial input data at time i. The deducing process can be indicated by the following relational expression.

$$crc[n-1]_i = crc[n-2]_{i-1} \oplus g[n-1]\times(d_{i-1} \oplus crc[n-1]_{i-1}) \quad (1)$$

$$crc[n-2]_i = crc[n-3]_{i-1} \oplus g[n-2]\times(d_{i-1} \oplus crc[n-1]_{i-1}) \quad (2)$$

……………………………..

$$crc[1]_i = crc[0]_{i-1} \oplus g[1]\times(d_{i-1} \oplus crc[n-1]_{i-1}) \quad (3)$$

$$crc[0]_i = d_{i-1} \oplus crc[n-1]_{i-1} \quad (4)$$

According the table1, the polynomial of CRC 8 can be expressed as G(x) $x^8+x^5+x^4+1$. The above equation (1)-(4) can be expressed by equation (5) and equation (6) by the way of matrix.

$$\begin{bmatrix} crc[7]_i \\ crc[6]_i \\ crc[5]_i \\ crc[4]_i \\ crc[3]_i \\ crc[2]_i \\ crc[1]_i \\ crc[0]_i \end{bmatrix} = \begin{bmatrix} 01000000 \\ 00100000 \\ 10010000 \\ 10001000 \\ 00000100 \\ 00000010 \\ 00000001 \\ 10000000 \end{bmatrix} \begin{bmatrix} crc[7]_{i-1} \\ crc[6]_{i-1} \\ crc[5]_{i-1} \\ crc[4]_{i-1} \\ crc[3]_{i-1} \\ crc[2]_{i-1} \\ crc[1]_{i-1} \\ crc[0]_{i-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ d_{i-1} \\ d_{i-1} \\ 0 \\ 0 \\ 0 \\ d_{i-1} \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} crc[7]_{i+1} \\ crc[6]_{i+1} \\ crc[5]_{i+1} \\ crc[4]_{i+1} \\ crc[3]_{i+1} \\ crc[2]_{i+1} \\ crc[1]_{i+1} \\ crc[0]_{i+1} \end{bmatrix} = \begin{bmatrix} 01000000 \\ 00100000 \\ 10010000 \\ 10001000 \\ 00000100 \\ 00000010 \\ 00000001 \\ 10000000 \end{bmatrix} \begin{bmatrix} crc[7]_i \\ crc[6]_i \\ crc[5]_i \\ crc[4]_i \\ crc[3]_i \\ crc[2]_i \\ crc[1]_i \\ crc[0]_i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ d_i \\ d_i \\ 0 \\ 0 \\ 0 \\ d_i \end{bmatrix} \quad (6)$$

According to the (1)-(6) equation, the relationship can be deduced among $crc[n-k]_i$,$crc[n-k-1]_{i-1}$ and d_{i-1} , in which the value of k ranges from 0 to n-1. These polynomials can be calculated any time by the above equations. For the m-bit parallel data, we can assume that only 1 bit can come in at each time from input port. So it will take m clock to calculate the m-bit data. These data will be calculated by mode 2 division with CRC polynomial and shift toward right 1 bit as the result of parallel CRC coding. According to the above (1)-(6) logical relation equation, we can eventually get the relationship among $crc[n-k-1]_{i+n-1}$, $crc[n-k-1]_i$ and input data d_i, d_{i+1} …… d_{i+m-1} after m clock. In order to calculate the CRC algorithm by matrix, we can assume the characteristic of matrix A.

$$A = \begin{bmatrix} 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 1&0&0&1&0&0&0&0 \\ 1&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \end{bmatrix}$$

According Figure 1, we assume Y(t) as the output value of the D flip-flop at time t. We assume d(t) as external input signal depending on the calculation structure of classic serial method of CRC8. It can be presented as shown in following at time t+1 and t+2.

$$Y(t+1) = A[Y(t)+d(t)] = AY(t) + A\begin{bmatrix} d(t) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$Y(t+2)=A[Y(t+1)+d(t+1)]=A\{A[Y(t)+d(t)]+d(t+1)\}=A^2Y(t)+A^2d(t)+Ad(t+1)$$

According to A matrix, it can be found that it is similar with the triangle diagonal matrix. Starting from the second column, each column only has one element with value 1 and the value of the rest element is 0. When the matrix takes iterative computation, it is easy to compute by such characteristics. Such as, matrix A multiply matrix A, we can find the regular pattern that first column will take multiplication and other column will move to the next column along the right direction as the new column. According the above characteristic, it will restructure the input data di and converted it into 1 column 8 structure for operation, which is very convenient for matrix iterative multiplication more times. According to the characteristics of mode 2 operation, if the element value is an even number in the matrix, it can be considered as 0. If the element value is an odd number can be considered as 1. So y (t+2) can be simplified as the following.

$$Y(t+2)=A^2Y(t)+A^2d(t)+Ad(t+1)=A^2Y(t)+A\begin{bmatrix}d(t+1)\\d(t)\\0\\0\\0\\0\\0\\0\end{bmatrix}$$

The structure of serial CRC 8 only has registers, so the matrix only can be iterated 8 times. At time t+8, the expression of Y(t +8) can be simplified as following:

$$Y(t+8)=A^8Y(t)+A\begin{bmatrix}d(t+8)\\d(t+7)\\d(t+6)\\d(t+5)\\d(t+4)\\d(t+3)\\d(t+2)\\d(t+1)\\d(t)\end{bmatrix}$$

Because the polynomial only has eight DFF, It can not be iterated at time t+9 according the above method. It has to restart by adding a iteration, as shown below:

$$Y(t+9)=A[Y(t+8)+d(t+9)]=A^9Y(t)+A^2\begin{bmatrix}d(t+8)\\d(t+7)\\d(t+6)\\d(t+5)\\d(t+4)\\d(t+3)\\d(t+2)\\d(t+1)\\d(t)\end{bmatrix}+A\begin{bmatrix}d(t+9)\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}$$

And The Like, for 64-bit data in parallel, the result of iterating 64 times is shown in following:

$$Y=A^{64}\begin{bmatrix}d1\\d2\\d3\\d4\\d5\\d6\\d7\\d8\end{bmatrix}+A^{56}\begin{bmatrix}d9\\d10\\d11\\d12\\d13\\d14\\d15\\d16\end{bmatrix}+A^{48}\begin{bmatrix}d17\\d18\\d19\\d20\\d21\\d22\\d23\\d24\end{bmatrix}+A^{40}\begin{bmatrix}d25\\d26\\d27\\d28\\d29\\d30\\d31\\d32\end{bmatrix}+A^{32}\begin{bmatrix}d33\\d34\\d35\\d36\\d37\\d38\\d39\\d40\end{bmatrix}+A^{24}\begin{bmatrix}d41\\d42\\d43\\d44\\d45\\d46\\d47\\d48\end{bmatrix}+A^{16}\begin{bmatrix}d49\\d50\\d51\\d52\\d53\\d54\\d55\\d56\end{bmatrix}+A^{8}\begin{bmatrix}d57\\d58\\d59\\d60\\d61\\d62\\d63\\d64\end{bmatrix}$$

In order to find the Y values, we only need to calculate the matrix value of A^{64} A^{56} A^{48} A^{40} A^{32} A^{24} , A^{16} A^{8}. These operations can be implemented by MATLAB tool. By more times matrix A multiplying matrix A , Y(t) can be expressed by t 64-bit input data multiplying new matrix . So we can gen the CRC checking code by the above matrix calculation for input 64- bit data. If the bit number of input data is not 8 multiple, we should add 0 as the input data at the end of input data to ensure that the number of input data is 8 multiple. By the method of matrix iteration, we can realize the 64 bit parallel CRC algorithm for high-speed data checking.

IV. IMPROVED CRC ALGORITHM BY PIPELINE TECHNOLOGY

According to the above content, we can get the parallel CRC algorithm by the matrix transformation[3,4] and derivation. For hardware, it can be realized by the combination circuit. Due to the whole of system, it has higher timing demand for system. So a large number of combination logic circuit has become the bottleneck in the system for high speed transmitting system. In order to overcome the timing restriction of the existing algorithm, we increase pipeline technology based on above algorithm, which can reduce the timing influence of the module in the system. The above relevant combinational circuit module has been realized by matrix transformation, which can be divided a number of little pattern. The little pattern can be connected with input and output by a number of D flip-flop, as shown in figure 2.

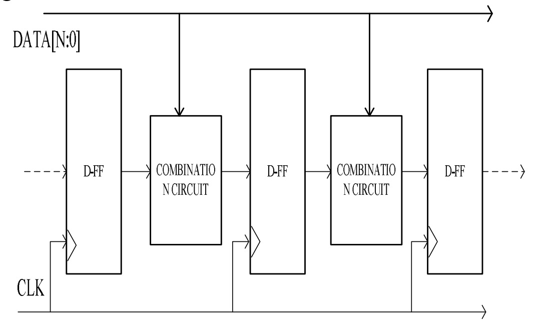

Figure 2 circuit structure of pipeline CRC

If we do not use pipeline technology, 64 bit parallel CRC algorithm will need to delay time x. If we use the 4 level pipeline technology algorithm[5], the complexity of combination circuit will be reduced to 1/4.It only needs a clock cycle to realize put the 64 bit parallel data calculation. The delay time will be reduced to the quarter of original The throughput rate data will raise 4 times. The whole design has been implemented by FPGA with VERILOG[6,7].

V. THE CONCLUSION

Due to the development of the high-speed network and data compression storage technology, the CRC technology work as a traditional calibration method, which has applied in a large number of communication protocol. Because of the whole system processing speed more and more quickly, the hardware circuit require the less and less time delay in system. So we must design CRC algorithm and the circuit structure with high speed and low delay time[8,9]. This article is built on the classic serial CRC coding mode and realization structure, which describe the new way to realize parallel CRC coding by matrix transformation.It reduces the delay time and improves the data throughput rate.

REFERENCES

[1] Pretzel O. Error-Correcting Code and Finite fields.[J].Oxford: Oxford University Press ,1992

[2] Albertengo G, Sisto R. Parallel CRC generation[J]. IEEE Micro 1990,10(5):58-80

[3] Derby J H. High-speed CRC computation using state space transformations[J]. In Proceedings of GLOBECOM’01,San Antonio,TX,USA,2001.158-175.

[4] Peterson and Wdldon, Error-Correcting Codes[M]. The MIT Press, 2nd edition 1972.

[5] Glaise R J. A two-step computation of cyclic redundancy code CRC-32 for ATM networks[J]. IBM Journal of Reserarch and development, 1997, 41(6): 700-710.

[6] Janick Bergeron. Writing Testbencher:Functional Verification of HDL Models Third Edition.BOSTON[M]:Kluwer Academic Pulishers,2003

[7] Samir Palntikar. Verilog HDL-A Guide to Digital Design and Synthesis[M]. Sunsofe Press, 1996, page:6

[8] Andrew Piziali.Functional Verification Coverage Measurement And Analysis. Boston[M] : Kluwer Aeademie Publishers, 2004

[9] Donalda E.Thomas;Philip R.Moorby.The Verilog Hardware Deschiption language[M].Lkuwer Academic Publishers,2002Article in a conference proceedings: