

A Method Of Arrangement Of An Algorithm In Cyclic Redundancy Check

FIELD OF THE INVENTION

The present invention pertains to the field of error correction in digital data processing and, more particularly, to an algorithm for generating system to calculate CRC
5 independent of the length of polynomial generator and data stream.

BACKGROUND OF THE INVENTION

Error correction techniques and architectures are well known in digital data processing and communications systems, including systems having data storage subsystems such as magnetic, optical, or semiconductor based memory stores. Detection and, where
10 possible, correction of erroneous data has been achieved by using an encoder circuit to construct some number of "redundant" m-bit error check symbol, which mathematically characterize the information in a selected block of data. The error check symbol is then appended to the data block and transferred through a communication channel. When the data block is received, or later retrieved from memory, the accuracy or reliability of
15 the data can be evaluated by use of these appended error check symbol.

Cyclic Redundancy check (CRC) is a classical error detection mechanism that is widely known in the art and used in numerous applications as a means to detect errors from a transmitted stream of information. Its general acceptance is due to the simplicity of implementation by which it can be used to encode into a stream of information to be
20 transmitted and subsequently checked at the receiver.

CRC calculation is performed in polynomial arithmetic, whereby the basis of the algorithm is; that given both transmitter and receiver mutually acknowledge the use of the same polynomial generator, which is essentially a sequence of information used to construct and verify the error check symbol (in the context of CRC, it is also known as
25 checksum) from the data stream, a stream of information (also known as packets or frames) is transmitted, whereby calculation of the error check symbol is created by performing polynomial division on the data stream against the chosen polynomial generator.

In prior art, in order to optimize the speed at which the error check symbol can be
30 calculated, some have opted for the solution of deriving Boolean or logical expressions that are valid for specific polynomial generators with fixed data width, while other

methodologies optimize the idea of using LFSR (linear feedback shift registers), but instead of single bit shifts, many bits or bytes may be shifted per clock cycle, however this then imposes additional rules for the algorithm. Further still, some methodologies proposed by the prior art attempts to optimize operation of performing CRC by checking for unchanged parts of certain streams of information, while others attempt to improve methodology by calculating partial remainders based on segments of the packet of information.

SUMMARY OF THE INVENTION

- 10 The present invention provides a method of arrangement of an algorithm to perform parallel cyclic redundancy check (CRC) comprising the step of: (i) considering a polynomial generator $g(x) = g_{k-1}x^{k-1} + g_{k-2}x^{k-2} + g_{k-3}x^{k-3} + \dots g_0$, and $f(x) = m(x)x^{k-1} + c(x)$; and $m(x) = m_{L-1}x^{L-1} + m_{L-2}x^{L-2} + m_{L-3}x^{L-3} + \dots m_0$; (ii) transforming $f(x)$ and $g(x)$ into a table as a function of each other; (iii) considering an exemplary case whereby $k=7$ and $L=7$, obtaining $f_{n-1} = A$, $f_{n-1} = A + B$, $f_{n-1} = A + B + C \dots f_2 = E + F + G$, $f_1 = F + G$, $f_0 = G$, wherein A, B, C.... G are divisor coefficient, and (iv) obtaining $c_{k-2} = f_{k-2}$, $c_{k-2} = f_{k-2} \dots c_1 = f_1$, $c_0 = f_0$.

It is an object of the present invention to provide a method of arrangement of an algorithm that calculate the checksum used in cyclic redundancy checks, which is generic and independent of the length of the polynomial generator and data stream, and the method can be realized in digital logic implementation limited only by the propagation delay of the slowest synthesized combinatorial logic path and typically calculated within a period of a single system clock cycle.

Another object of the present invention is to provide a method of arrangement of an algorithm that calculates the checksum used in cyclic redundancy check, wherein a sequence of information and the corresponding polynomial generator are transformed into a table to simplify the realization and implementation of the method in synthesizable logic.

These and other objects, aspects, advantages and features of the present invention will be more fully understood and appreciated by those skilled in the art upon consideration of the following detailed description of a preferred embodiment, presented in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

It is to be understood that the accompanying drawings are provided for the purpose of illustration only, and are not intended as a definition of the limits of the invention. The drawings illustrate a preferred embodiment of the present invention, in which:

5 FIG. 1 shows a transformation table embodying a 7 bit data stream and a 7 bit polynomial generator in accordance with the present invention;

FIG. 2 shows an exemplary CRC checksum calculation in accordance with the present invention; and

10 FIG. 3 shows USB CRC5 implementation as LFSR using generator polynomial $G(x) = x^5 + x^2 + 1$ of a conventional method.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

CRC calculation is performed in polynomial arithmetic, whereby the basis of an algorithm is given that both the transmitter and the receiver use the same polynomial generator sequence, before a stream of information (also known as packets or frames) is transmitted, calculation will be made against the selected polynomial generator via polynomial division. The remainder of the result of the polynomial division is appended against the original packet of information to be transmitted. Thence when the information is transmitted and received by the receiver, a verification can be made by performing the same polynomial division to the received packet of information and checking if the resultant remainder is equal to zero, and hence an error is easily determined for cases when the remainder is not.

In the present invention, the method is employed to provide a method of arrangement of an algorithm that calculates the checksum used in CRC. With respect to mathematical context, the disclosure above can be explained by the following representation.

25 Suppose that the data stream is represented by $m(x)$ with coefficients $m_{L-1}, m_{L-2}, \dots$ and each polynomial degree is represented by x^{n-1} where n is the degree of the polynomial ranging from $\{0, n-1\}$, then the data stream can be said that

$$m(x) = m_{L-1}x^{L-1} + m_{L-2}x^{L-2} + m_{L-3}x^{L-3} + \dots m_0$$

30 Where L is the length of the data packet.

Additionally the polynomial generator can be similarly represented by

$$g(x) = g_{k-1}x^{k-1} + g_{k-2}x^{k-2} + g_{k-3}x^{k-3} + \dots g_0$$

Where k is the length of the polynomial generator. In the present invention, the CRC
5 checksum calculated can be represented by

$$c(x) = c_{k-2}x^{k-2} + c_{k-3}x^{k-3} + c_{k-4}x^{k-4} + \dots c_0$$

The final frame of information to be transmitted after CRC has been calculated and
appended to $m(L)$ can be represented as:

10
$$f(x) = m(x)x^{k-1} + c(x)$$

And let $q(x)$ be the quotient product of data stream (appended by $k-1$ bits) and the
polynomial generator and $m(x)$ can be rewritten as

$$m(x)x^{k-1} = g(x)q(x) + c(x)$$

15 Where the solution is derived from the following equation:

$$c(x) = \text{remainder}\left(\frac{m(x)x^{k-1}}{g(x)}\right)$$

In prior art, Figure 2 shows how $c(x)$ is realized, and it can be seen that the present
method of polynomial division infers a need to retain information about the past, in order
20 to bring it forward and eventually arrive at a resultant value by which is the remainder.
In digital logic systems, a number of Linear Feedback Shift Registers (LFSR) each
corresponding to a degree of the polynomial up to the top most degree associated to the
polynomial generator, are cascaded in series with additional Exclusive Or (XOR) logic
wherever the degree of the polynomial generator is 1. An example of which can be seen
25 in Figure 3.

As a result, it would require minimum of $k-1$ number of clock cycles (where $k-1$ is the
highest degree associated with the polynomial generator) in order to calculate the CRC
for a given transmitted stream at the receiver side.

FIG. 1 illustrates an exemplary transformation of a 7 bit data stream and a 7 bit
30 polynomial generator to a table in accordance with the present invention. As shown, the
transformation table as described by this present invention of data stream plus CRC

check sum ($f(x)$: $f_0, f_1, f_2, f_3 \dots f_{10}, f_{11}, f_{12}$), against the polynomial generator sequence ($g(x)$: $g_0, g_1 \dots g_5, g_6$), giving rise to the divisor coefficients A, B... F, G. The divisor coefficients are used to calculate the CRC sequence in accordance with the present invention.

- 5 As shown in FIG. 2, the calculation of long division being performed on the data stream 1100100111 with additional 0000 being appended to the stream from the polynomial sequence 10101 or x^4+x^2+1 . In the present preferred embodiment, the remainder at the bottom replaces the additional 0000 appended to the data stream as the CRC checksum.

10 In the preferred embodiment of the present invention, in order to effectively work out the algorithm, $f(x)$ and $g(x)$ are transformed into a table as a function of each other as shown in FIG. 1. Taking an exemplary case of $L = 7$ and $k = 7$ and where the blank spaces are corollary null in value. The mathematical meaning of $q(x)$ becomes meaningless after making the transformation, and it can be seen that coefficients A, B, C ... F, G arise, and let these coefficients be named as divisor coefficients.

- 15 Considering a special case where the polynomial generator is such that all the coefficients from the k th degree 0 to 6 are all one, and let n be the length of the data stream plus the length of the CRC check sum ($L+(k-1)$), it can be derived that

$$f_{n-1} = A, f_{n-1} = A + B, f_{n-1} = A + B + C \dots f_2 = E + F + G, f_1 = F + G, f_0 = G$$

- 20 And therefore,

$$c_{k-2} = f_{k-2}, c_{k-2} = f_{k-2} \dots c_1 = f_1, c_0 = f_0$$

- 25 If the polynomial generator is not special case, then the derivation above would require an extra multiplication factor based on the row index of the polynomial generator as shown in FIG. 1, with the row of divisor coefficients that have been shifted.

30 In accordance with the present invention, the algorithm is being used as disclosed to generate a system for CRC checksum which is independent of data or polynomial generator length typically calculated within one clock cycle, limited only by the propagation delay of the slowest synthesized combinatorial logic path. The generation of the system determines the resultant reminder from performing polynomial division on a data sequence from a configurable polynomial generator without requiring knowledge about the past or having to implement feedback which implies memory. The present

invention provides a novel approach by determining the coefficients of the quotient of the result from performing polynomial division by assuming that the remainder is zero, which infers that the remaining $k-1$ bits of the overall quotient is what would be the remainder of the polynomial division.

- 5 Thus, it would be apparent to those skilled in the art that many more modifications are possible without departing from the inventive concepts herein. The invention, therefore, is not to be restricted except in the spirit of the appended claims.

What is claimed is:

1. A method of arrangement of an algorithm to perform parallel cyclic redundancy check (CRC) comprising the step of: (i) considering a polynomial generator $g(x) = g_{k-1}x^{k-1} + g_{k-2}x^{k-2} + g_{k-3}x^{k-3} + \dots + g_0$, and $f(x) = m(x)x^{k-1} + c(x)$; and $m(x) = m_{L-1}x^{L-1} + m_{L-2}x^{L-2} + m_{L-3}x^{L-3} + \dots + m_0$; (ii) transforming $f(x)$ and $g(x)$ into a table as a function of each other; (iii) considering exemplary case of $k = 7$ and $L = 7$, obtaining $f_{n-1} = A$, $f_{n-1} = A + B$, $f_{n-1} = A + B + C$... $f_2 = E + F + G$, $f_1 = F + G$, $f_0 = G$, wherein A, B, C... G are divisor coefficient, and (iv) obtaining $c_{k-2} = f_{k-2}$, $c_{k-2} = f_{k-2}$... $c_1 = f_1$, $c_0 = f_0$.
2. The method of arrangement of an algorithm of claim 1, wherein in step (iii), the derivation requires an extra multiplication factor based on the divisor coefficients of the polynomial generator, if all the coefficients of $g(x)$ of the polynomial generator is not equal to 1.
3. The method of arrangement of an algorithm of Claim 1, wherein the divisor coefficients, A, B ... F, G are used to obtained CRC checksum from the data stream against the polynomial generator sequence by utilization of the mathematical formulation that can be realized in step (iii) of Claim 1.
4. The method of arrangement of an algorithm of Claim 1, wherein the algorithm is independent of the length of the polynomial generator and data stream.
5. The method of arrangement of an algorithm of Claim 1, wherein the algorithm realized in digital logic implementation is limited in speed only by the propagation delay of the slowest synthesized combinatorial logic path with a typical latency of one system clock cycle.

A Method Of Arrangement Of An Algorithm In Cyclic Redundancy Check

Abstract of the disclosure

5 A method of arrangement of an algorithm to calculate cyclic redundancy check (CRC) independent of the length of a polynomial generator and data stream which can be realized in digital implementation with a calculation latency of once clock cycle. The method allows a sequence of information and the corresponding polynomial generator be arranged into a transformation table.

10 (Fig. 1)

	f ₁₂	f ₁₁	f ₁₀	f ₉	f ₈	f ₇	f ₆	f ₅	f ₄	f ₃	f ₂	f ₁	f ₀
g ₆	A	B	C	D	E	F	G						
g ₅		A	B	C	D	E	F	G					
g ₄			A	B	C	D	E	F	G				
g ₃				A	B	C	D	E	F	G			
g ₂					A	B	C	D	E	F	G		
g ₁						A	B	C	D	E	F	G	
g ₀							A	B	C	D	E	F	G

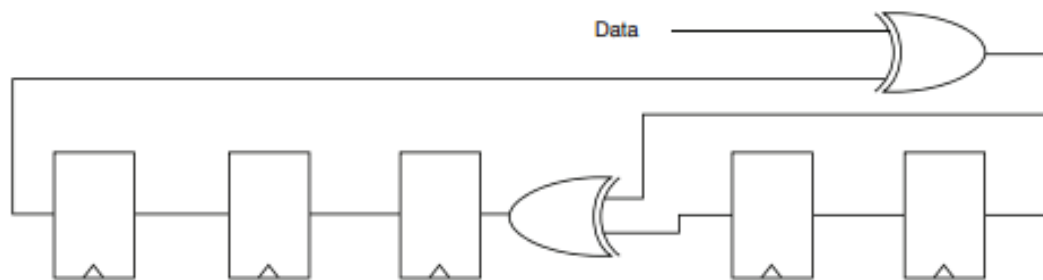
FIG. 1

```

10101}11001001110000(1111100001
  10101}11111
  11000}11111
  10101}11111
  11010}11111
  10101}11111
  11111}11111
  10101}11111
  10101}11111
  10101}11111
  00001}11111
  00000}11111
  00010}11111
  00000}11111
  00100}11111
  00000}11111
  01000}11111
  00000}11111
  10000}11111
  10101}11111
  0101}11111

```

FIG. 2



PRIOR ART

FIG. 3