Reprint

Single - Bit Error Correction Circuit for ATM Interfaces

A. Maniatopoulos, T. Antonakopoulos and V. Makios

Electronics Letters

VOL. 31 NO. 8, APRIL 1995, pp. 617-618

Single-bit error-correction circuit for ATM interfaces

A. Maniatopoulos, T. Antonakopoulos and V. Makios

Indexing terms: Asynchronous transfer mode, Error correction

Cell switching systems use cyclic codes for protecting cell headers from transmission errors either by detecting multiple errors or by correcting single-bit errors. The Letter presents a new method for implementing single-bit forward error-correction functions by minimising the complexity of parallel CRC circuits, resulting in low hardware complexity and high operational speed. The method does not use a look-up table for determining the corrupted bit position, but implements a repetitive algorithm for matching the generated syndrome. The implementation of the proposed method to an ATM interface using field programmable gate arrays is also described.

Introduction: A broadband integrated services digital network (B-ISDN) uses the asynchronous transfer mode (ATM) as the basic technique for information exchange. In ATM, fixed-length cells are transmitted consisting of a header and a payload [1]. An error-correcting code is used in the cell header for achieving enhanced transmission reliability, since the cell header supports network routing functions. The error-correction function is based on cyclic code theory and is usually implemented using a look-up table, which contains the syndrome values [2].

Although the cyclic redundancy code (CRC) calculation is a well documented process, the determination of the position of the corrupted bit in the cell header is a laborious task and affects the circuit complexity and its maximum operational speed. In this Letter, a new method for single-bit error-correction is presented. The analysis of the proposed method is given initially, concentrating on parallel data formats, and its implementation in an ATM interface is described.

Implementation constraints: When (m,n) bits of data are transmitted using the CRC method, codewords, represented as $M(x) = \Sigma_{i=0}^{m-1} m_i x^i$, are produced using the $G(x) = \Sigma_{i=0}^{n} g_i x^i$ generator polynomial. During codeword transmission, if a single bit error is inserted, the received message is the sum of the transmitted message $M(x)$ and of a term x^i, where i is a number between 0 and $(m-1)$, indicating the position of the corrupted bit. Since $M(x)$ is a codeword derived by applying the CRC algorithm using the $G(x)$ generator polynomial, the remainder left over from dividing $x^n M(x)$ by $G(x)$, denoted as $R_{G(x)}\{x^n M(x)\}$ [3], is equal to zero in error-free transmissions. So, the remainder resulting at the receiver from the division of $x^n(M(x)+x^i)$ by $G(x)$ is

$$R_{G(x)}\{x^n(M(x)+x^i)\} = R_{g(x)}\{x^n M(x)\} + R_{G(x)}\{x^n x^i\} = R_{G(x)}\{x^n x^i\} \quad (1)$$

since the remainder of a sum of polynomials under division by another polynomial is the sum of the remainders of the individual polynomials. Generally, in order to implement single-bit error-correction functions, the m possible patterns $R_{G(x)}\{x^n x^i\}$ are stored in a look-up table and the position of the corrupted bit in the received message is determined by comparing the division remainder with the look-up table content. The correction is performed by inverting the bit at the respective position.

The use of a look-up table cannot be considered as the optimum solution, since a part of the available silicon area is wasted for storing this table. In some applications, the use of a look-up table is impractical, due to constraints imposed by the technology used and by the cost of additional circuitry. In particular, when FPGAs are used, the use even of a small look-up table requires a large portion of the available functional blocks. In some cases, even for ASIC developments, as in [4], the error-correction function is not implemented, based on the assumption that the channel has a small BER.

This Letter presents a new method for implementing single-bit error-correction functions without using a look-up table. The method uses a repetitive procedure for determining the position of the erroneous bit. It has been implemented using FPGAs in high circuit speeds, where the best utilisation of the chip resources is critical and there are limited capabilities for implementing memory cells. Although the method can be analysed and implemented using either serial or parallel data representation, the analysis that follows is based on the parallel data format, since it was initially developed for STM interfaces, which exploit the SDH framing structure for achieving byte-level synchronisation.

Method analysis: In the parallel data format, the received codeword consists of a number of data blocks, where each block contains λ bits, and the total number of blocks of each codeword is equal to $b = m/\lambda$. The hardware implementation of the parallel CRC computation consists of an XOR-plane which combines the current data block with the syndrome computed from the preceding data blocks. For determining the XOR-plane in the general case, the following state transition equation is applied, which combines λ shift and conditional subtract operations within a single clock cycle [5]:

$$R(t+\lambda) = [r_0\, r_1 \ldots r_{n-\lambda-1}][O|I_{n-\lambda}] \oplus ([r_{n-\lambda}\, r_{n-\lambda+1} \ldots r_{n-1}] \oplus \mathbf{Z}(t))\mathbf{D} \quad (2)$$

where $[r_0\, r_1 \ldots r_{n-1}]$ is the syndrome at the t-transition, and $Z(t) = [z_t\, z_{t+1} \ldots z_{t+\lambda-1}]$ is the vector containing a block of λ bits, and

$$\mathbf{D} = \begin{bmatrix} \mathbf{G} \\ \mathbf{GT}^1 \\ \vdots \\ \mathbf{GT}^{i-1} \end{bmatrix} \quad \text{with } \mathbf{T} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ g_0 & g_1 & g_2 & \cdots & g_{n-1} \end{bmatrix}$$

The m syndrome patterns of eqn. 1 are represented by a $(b \times \lambda)$ matrix $\mathbf{S}$:

$$\mathbf{S} = \begin{bmatrix} s_{0,0} & s_{0,1} & \cdots & s_{0,\lambda-1} \\ s_{1,0} & s_{1,1} & \cdots & s_{1,\lambda-1} \\ \cdots & \cdots & \cdots & \cdots \\ s_{b-1,0} & s_{b-1,1} & \cdots & s_{b-1,\lambda-1} \end{bmatrix}$$

where

$$s_{ij} = R_{G(x)}\{x^n x^{\lambda i+j}\} \quad 0 \le i \le b-1 \quad 0 \le j \le \lambda-1$$
$$s_{b-1,\lambda-1} = R_{G(x)}\{x^n x^{m-1}\} \quad i = b-1, j = \lambda-1 \quad (3)$$

The last row of matrix $\mathbf{S}$ contains the syndromes of all the single bit errors occurring in the first incoming block of the received message, while the last column consists of the syndromes of all single-bit errors occurring in the most significant bit of each block of the received message. Assuming that the message has been corrupted by an error at the i-bit of the j-block, the remainder left corresponds to the s_{ij} pattern. Applying eqn. 2, after $(b-1-i)$ cyclic shifts through the XOR-plane, the result left in the CRC circuit flip-flops is the $s_{b-1,i}$ element of the syndrome matrix. In this case, an appropriately defined AND-plane of λ n-input gates can be used for single bit error detection, since the AND-plane is capable of detecting all single bit errors occurring in the first block of the message. The AND-plane is given by

$$e_{i-(m-\lambda-1)} = R_{G(x)}\{x^n x^i\} \text{ for } m-\lambda-1 \le i < m \quad (4)$$

The number of shift cycles is measured by a modulo-b counter which indicates the block of the message where the error occurred. If the initial data have been stored in a $(b \times \lambda)$ first-in-first-out memory, the corrupted bit can be corrected during memory read cycles using an XOR-plane, which is activated when the respective data block passes through. Fig. 1 shows the circuit that implements this method.

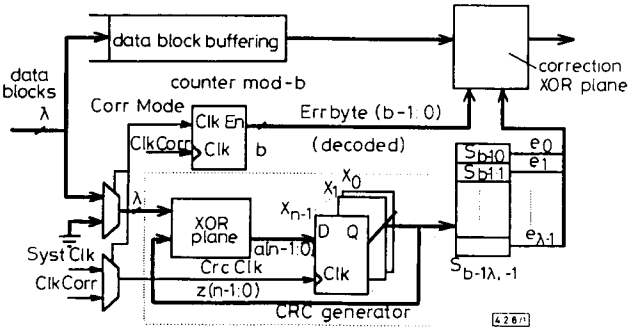

Fig. 1 *CRC circuit for single-bit error-correction in parallel data formats*

Application of method in ATM interfaces: The above method has been used in the STM-1 interface of B-ISDN. Owing to the frame structure of the SDH transmission interface, byte synchronisation is achieved at the SDH framing circuit and a byte stream is passed to the ATM cell-level based functions. The fifth byte of the header of an ATM cell is the CRC of the first four bytes containing the routing information for the cell. The generator polynomial used for the syndrome generation is the CCITT CRC-8 polynomial $G(x) = 1+x+x^2+x^8$ [6]. If the cell header is processed a byte at a time, the parameters of eqn. 2 become $\lambda = 8$, $n = 8$ and $m = 40$, and the equation is simplified to

$$R(t+8) = ([r_0\, r_1 \ldots r_l] \oplus \mathbf{Z}(t))\mathbf{D} \qquad (5)$$

After some mathematical manipulations, the matrix $\mathbf{D}$ is calculated to be equal to

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (6)$$

while the XOR and the AND planes, presented in Table 1, are derived by using eqns. 4 and 5.

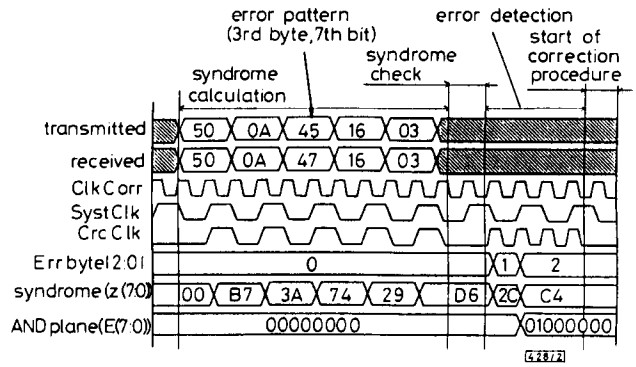

Fig. 2 *Circuit timing diagram with single-bit error-correction function*

The circuit of Fig. 1 can be used for error detection and correction for ATM interfaces, like STM-1, if 8 bits are considered in each data block. The part that is contained in the dashed rectangle can be used by the transmitter for generating the HEC byte of the ATM cell header. Fig. 2 shows the circuit operation when the received data contain a single error in the 22nd position (byte 3, bit 6). The correction circuit is activated when a nonzero syndrome is detected and requires four cycles for determining the corrupted bit position. The CRC calculation clock is equal to the data rate, while the correction clock can be much higher and depends only on the total circuit propagation delay. When the error position has been determined, the content of the AND-plane and the values of the counter are frozen, up to the end of the correction procedure. The CRC generation and correction circuit has been implemented in an XC3164 FPGA, requiring 20 CLBs, six CLBs for the CRC generator and 14 for the correction part. It has been tested at 19.44MHz (155.52Mbit/s) in an STM-1 SDH interface, at 77.76MHz (622Mbit/s) for the STM-4 interface, while its maximum speed for this type of FPGA has been measured to be 90MHz by optimising the placement of the CLBs.

Table 1: XOR and AND plane functions for CCITT CRC-8 polynomial

XOR plane	AND plane
$a_0 = z_0 \oplus r_0 \oplus z_6 \oplus r_6 \oplus z_7 \oplus r_7$	$e_0 = \bar{q}_7 \cdot q_6 \cdot q_5 \cdot \bar{q}_4 \cdot \bar{q}_3 \cdot \bar{q}_2 \cdot q_1 \cdot \bar{q}_0$
$a_1 = z_0 \oplus r_0 \oplus z_{1\S} \oplus r_1 \oplus z_6 \oplus r_6$	$e_1 = q_7 \cdot q_6 \cdot \bar{q}_5 \cdot \bar{q}_4 \cdot \bar{q}_3 \cdot q_2 \cdot \bar{q}_1 \cdot \bar{q}_0$
$a_2 = z_0 \oplus r_0 \oplus z_1 \oplus r_1 \oplus z_2 \oplus r_2 \oplus z_6 \oplus r_6$	$e_2 = q_7 \cdot \bar{q}_6 \cdot \bar{q}_5 \cdot \bar{q}_4 \cdot q_3 \cdot q_2 \cdot q_1 \cdot q_0$
$a_3 = z_1 \oplus r_1 \oplus z_2 \oplus r_2 \oplus z_3 \oplus r_3 \oplus z_7 \oplus r_7$	$e_3 = \bar{q}_7 \cdot \bar{q}_6 \cdot \bar{q}_5 \cdot q_4 \cdot \bar{q}_3 \cdot q_2 \cdot q_1 \cdot \bar{q}_0$
$a_4 = z_2 \oplus r_2 \oplus z_{3\S} \oplus r_3 \oplus z_4 \oplus r_4$	$e_4 = \bar{q}_7 \cdot \bar{q}_6 \cdot q_5 \cdot q_4 \cdot \bar{q}_3 \cdot \bar{q}_2 \cdot q_1 \cdot \bar{q}_0$
$a_5 = z_3 \oplus r_3 \oplus z_{4\S} \oplus r_4 \oplus z_5 \oplus r_5$	$e_5 = \bar{q}_7 \cdot q_6 \cdot q_5 \cdot \bar{q}_4 \cdot \bar{q}_3 \cdot q_2 \cdot \bar{q}_1 \cdot \bar{q}_0$
$a_6 = z_4 \oplus r_4 \oplus z_{5\S} \oplus r_5 \oplus z_6 \oplus r_6$	$e_6 = q_7 \cdot q_6 \cdot \bar{q}_5 \cdot \bar{q}_4 \cdot q_3 \cdot \bar{q}_2 \cdot \bar{q}_1 \cdot \bar{q}_0$
$a_7 = z_5 \oplus r_5 \oplus z_{6\S} \oplus r_6 \oplus z_7 \oplus r_7$	$e_7 = q_7 \cdot \bar{q}_6 \cdot \bar{q}_5 \cdot q_4 \cdot \bar{q}_3 \cdot q_2 \cdot q_1 \cdot q_0$

Conclusions: This Letter presented a new method and the respective circuit for single-bit error correction in ATM interfaces. The method can be applied to any length of coded data and is independent of the CRC generating polynomial. The main advantages of the proposed circuit are its ability to determine the position of the corrupted bit without using any look-up table and its implementability using FPGAs.

Acknowledgments: The authors wish to thank the European Union for financially supporting this work as part of the RACE II project R2045 - DISTIMA.

28 February 1995

Electronics Letters Online No: 19950445

A. Maniatopoulos, T. Antonakopoulos and V. Makios (*Electrical Engineering Department, University of Patras, 26500 Patras, Greece*)

References

1 DRAVIDA, S., and DAMODARAM, R.: 'Error detection and correction options for data services in B-ISDN', *IEEE J. Sel. Areas Commun.*, 1991, **9**, (9), pp. 1484–1495

2 JOHNSTON, C.A., and CHAO, H.J.: 'The ATM layer chip: An ASIC for B-ISDN applications', *IEEE J. Sel. Areas Commun.*, 1991, **9**, (5), pp. 741–750

3 RAMABADRAN, T., and GAITONDE, S.: 'A tutorial on CRC computations', *IEEE Micro*, 1988, **8**, (4), pp. 62–75

4 TRAW, C.B.S., and SMITH, J.M.: 'Hardware/software organization of a high-performance ATM host interface', *IEEE J. Sel. Areas Commun.*, 1993, **11**, (2), pp. 240–253

5 PEI, T.B., and ZUKOWSKI, C.: 'High-speed parallel circuits in VLSI', *IEEE Trans.*, 1992, **COM-40**, (4), pp. 653–657

6 CCITT Recommendation I.432: 'B-ISDN user-network interface - Physical layer specification', Geneva, June 1992