Reprint

Implementation Issues of the ATM cell delineation mechanism

A. Maniatopoulos, T. Antonakopoulos and V. Makios

Electronics Letters

VOL. 32 NO. 1, MAY 1996, pp. 963-964

Implementation issues of the ATM cell delineation mechansim

A. Maniatopoulos, T. Antonakopoulos and V. Makios

Indexing terms: Asynchronous transfer mode, Synchronisation

The cell delineation mechanism (CDM) used in the asynchronous transfer mode (ATM) interfaces is based on the validation of the header error control (HEC) byte of the incoming cells, and is used for extracting the boundaries of the incoming cell stream. The CDM is analysed and an efficient implementation algorithm is derived, resulting in a hardware architecture, applicable at high data rates.

Introduction: The CDM is used to achieve synchronisation at any ATM interface, either using a framed (like SONET) or an unframed (like pure ATM) data transmission format. This mechanism exploits the correct HEC byte of any uncorrupted cell received from the network and is accomplished within two steps. Initially, the receiver is in the HUNT state and searches for a valid header codeword. Upon finding the first valid HEC, the receiver enters the PRESYNC state and the CDM mechanism is applied every 53 bytes. After six consecutive validations, the system goes into the SYNC state; otherwise, if an invalid HEC is detected within this period, the system returns to the HUNT state. In the SYNC and PRESYNC states, the accuracy of a cell header is often checked using a parallel CRC circuit based on the polynomial $G(x) = x^0 + x^1 + x^2 + x^8$ and by comparing the remainder resulting from the consecutive divisions with a predefined value [1]. This circuit is reset before the appearance of the first byte of each header and is enabled for five byte periods.

However, in the HUNT state, the same circuit cannot be used, since it must not be reset or disabled until a correct cell header indication is given. Thus, all methods compute the syndrome of the last five incoming bytes (or the last 40 bits) without requiring a system reset or enable signal, unless cell delineation is related to the cell mapping into a SONET frame [2]. When the CDM is not related to the data framing format, the syndrome is commonly computed by applying a sliding window over the incoming data stream [3]. The proposed mechanism presented in this Letter follows an accumulative procedure for computing two different remainders, which are then substracted (mod-2 addition) to produce the syndrome. The advantages of this approach over the sliding window method, along with the respective implementations, are referred to in the last Section.

Method description: In the HUNT state, the system begins to search for a valid HEC by calculating the syndrome for the last five incoming bytes. When the result is equal to a predefined value, these five bytes are considered to compose a valid header codeword and the system enters the PRESYNC state. Let $(n+5)$ denote the number of bytes that have been received from the beginning of the search procedure until the detection of the valid HEC. This information can be represented as a polynomial:

$$I(x) = M(x) + H(x) \qquad (1)$$

where $H(x) = h_0 x^0 + h_1 x^1 + \cdots + h_{39} x^{39}$ (the last 40 bits) and $M(x) = m_{40} x^{40} + m_{41} x^{41} + \cdots m_{8(N+5)-1} x^{8(n+5)-1}$ $n \geq 1$. Although the following analysis is focused on byte-parallel interfaces where byte boundaries are derived from a frame structure, it can also be applied to serial interfaces like the pure ATM, where cell delineation also results in the reconstruction of the byte stream. In this case, the respective equations which describe the system architecture can be derived by the general form of the well studied state transition equation of [4] when applied to a serial bit stream. We generally consider that the incoming message $I(x)$ consists of a number of information blocks which arrive successively in the HEC search machine. In byte-oriented systems, the procedure for the calculation of the remainder of the division of the message $x^k I(x)$, with a vector $G(x)$ of degree k, is based on the following equation:

$$R(t+8) = ([r_0, r_1, ..., r_7] + Z(t)) \cdot \mathbf{D} \qquad (2)$$

where $[r_0, r_1, ..., r_7]$ is the syndrome at the t-transition, and $Z(t) = [z_t, z_{t+1}, ..., z_{t+7}]$ is the vector of the incoming byte at the t-transition.

The '+' symbol here denotes XOR operation (modulo-2 addition). The **D** matrix for the $G(x) = x^0 + x^1 + x^2 + x^8$ generator polynomial has been calculated in [5] and is equal to

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

Following the representation of [6], we denote $R_{G(x)}\{M(x)\}$ the remainder of the division of a message $M(x)$ with a vector $G(x)$. In the proposed method two state transition machines are used in parallel to compute the remainders $R_{G(x)}\{x^8 I(x)\}$ and $R_{G(x)}\{x^8 M(x)\}$, so that the syndrome can be derived by the modulo 2 addition:

$$R_{G(x)}\{x^8 I(x)\} = R_{G(x)}\{x^8 (H(x) + M(x))\} + R_{G(x)}\{x^8 M(x)\} \qquad (4)$$

The first part of the sum of eqn. 4 is computed directly from eqn. 2 while the second one is calculated by modifying the state transition equation. The vector $Z(t)$, which represents the byte that enters the state machine at time t, must now be replaced by the term

$$Z'(t) = Z(t-5) \cdot \mathbf{D}^5 \qquad (5)$$

which denotes the remainder of the byte that has arrived five cycles ago. The term $Z'(t)$ can be derived either progressively while the $Z(t-5)$ byte is forwarded through a delay buffer or directly by computing the $\mathbf{D}^5$ matrix. These approaches are illustrated in Fig. 1*a* and *b*, respectively, and the benefits of each one are examined along with the evaluation of the proposed mechanism. The equations for the XOR planes are derived from the columns of **D** and $\mathbf{D}^5$ arrays and are given in Table 1. The block diagram of the proposed algorithm is shown in Fig. 2 where the direct calculation of $Z'(t)$ is depicted.

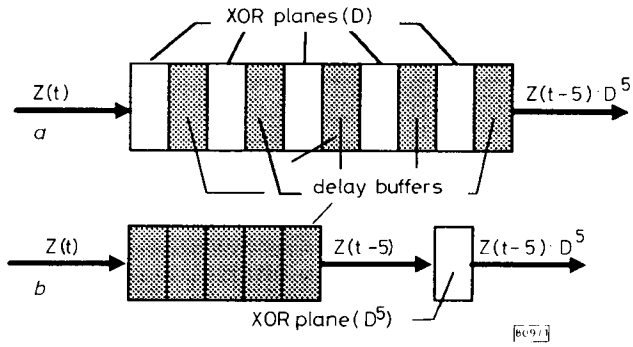

Fig. 1 *Calculation of Z'(t)*
a Direct, *b* progressive

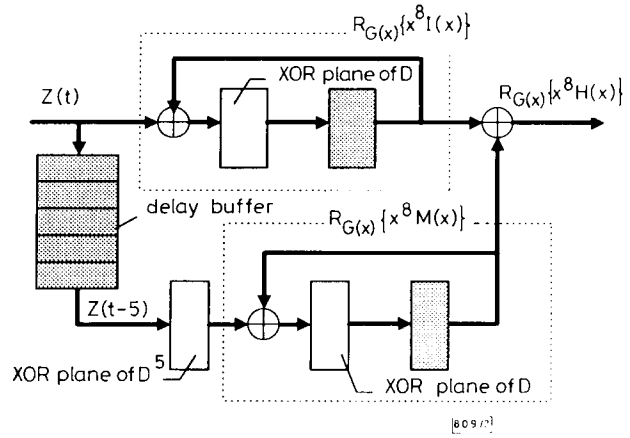

Fig. 2 *Cell delineation block diagram*

Table 1: Equations of XOR-planes derived from the columns of the arrays **D** and $\mathbf{D}^5$

	XOR plane from **D**	XOR plane from $\mathbf{D}^5$
r_0	$z_0 + z_6 + z_7$	$z_0 + z_3 + z_5$
r_1	$z_0 + z_1 + z_6$	$z_1 + z_3 + z_4 + z_5 + z_6$
r_2	$z_0 + z_1 + z_2 + z_6$	$z_2 + z_3 + z_4 + z_6 + z_7$
r_3	$z_1 + z_2 + z_3 + z_7$	$z_0 + z_3 + z_4 + z_5 + z_7$
r_4	$z_2 + z_3 + z_4$	$z_1 + z_4 + z_5 + z_6$
r_5	$z_3 + z_4 + z_5$	$z_0 + z_2 + z_5 + z_6 + z_7$
r_6	$z_4 + z_5 + z_6$	$z_1 + z_3 + z_6 + z_7$
r_7	$z_5 + z_6 + z_7$	$z_2 + z_4 + z_7$

Experimental results: The described CDM implementation has been used and tested in the STM-1 (155Mbit/s) and STM-4 (622Mbit/s) user network interfaces of B-ISDN. The implementation of this method is based on the XILINX XC3 100A series FPGAs and the maximum rate of operation that has been achieved is 80 MHz (640Mbit/s) when the progressive method is used. In lower speeds (like in the ATM interface at 25Mbit/s for desktop applications), the direct calculation method can reduce hardware complexity and can be further optimised by simplifying the combinatorial logic. Both of the above proposed methods have better performance when compared with other techniques that do not use state transition machines, like the one of Fig. 3 which implements the sliding window mechanism. The boxes that are labelled as powers of the matrix **D** are XOR-planes which are produced by the columns of these arrays. The delays that are imposed due to the large fan-out of the registers and the number of gates which compose the logic of these planes make this technique inapplicable in higher rates, especially when reconfigurable devices are used.

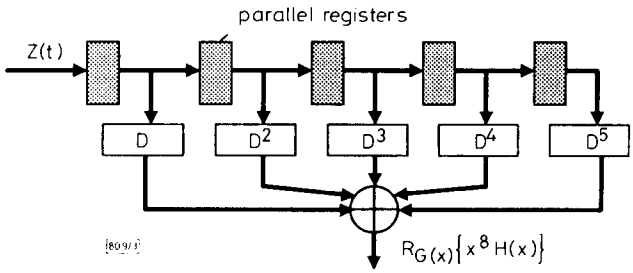

Fig. 3 *Syndrome calculation as sum of remainders*

Conlusions: In this Letter, an efficient implementation of the cell delineation mechanism, based on the HEC validation, has been described and the advantages of the implementation of this method in field programmable gate arrays were highlighted. Previous experience from implementations of STM-1 and STM-4 UNIs has shown that when this method is combined with other functions of the physical layer, like header error detection and single bit error correction [5], it gives optimum performance and minimises the hardware complexity.

Acknowledgment: This work was partially supported by the 'Research for Advanced Communications in Europe' (RACE) program, R2045 DISTIMA project.

25 March 1996

Electronics Letters Online No: 19960631

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

References

1 CCITT 1.432 Resolution 2, 1990, B-ISDN User-Network Interface — Physical Layer Specification

2 ATM Forum: 'Intermediate ATM: SONET cell delineation' Educational package, Release 1.0, August 1994

3 ROBE, T.J., and WALSH, K.A.: 'A SONET STS-3c user network interface integrated circuit', *IEEE J. Sel. Areas Commun.*, 1991, **9**, (5), pp. 732–740

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

5 MANIATOPOULOS, A., ANTONAKOPOULOS, T., and MAKIOS, V.: 'Single-bit error correction circuit for ATM interfaces', *Electron. Lett.*, 1995, **31**, (8), pp. 617–618

6 RAMABADRAN, , and GAITONDE, S.S.: 'A tutorial on CRC computations', *IEEE Micro*, 1988, **8**, (4), pp. 62–74