Reprint

Design and Implementation of a new Synchronization Method for High-Speed Cell-based Network Interfaces

A. Maniatopoulos, T. Antonakopoulos and V. Makios

The IEEE International Conference on Computer Design - ICCD'96

AUSTIN, TX, OCTOBER 1996

Design and Implementation of a new Synchronization Method for High-Speed Cell-based Network Interfaces

A. Maniatopoulos, T. Antonakopoulos and V. Makios
Department of Electrical Engineering and Computers Technology
University of Patras, 26500 Patras, Greece
Tel: +30 (61) 997 286, Fax: +30 (61) 997 342, e-mail: antonako@ee.upatras.gr

Abstract: *This paper presents the coding and timing recovery issues of an interface for high-speed cell-based communication networks. The synchronization technique used in this interface is based on the functionality resulting from combining functions of Fibre Channel with the basic features of pure ATM. The line coding properties, along with a pseudo-framing structure, are used to simplify the cell delineation process, thus resulting to lower hardware complexity than existing ATM interfaces, and implementability at high transmission rates. The paper describes also the architecture of a prototyped interface and its implementation using commercially available and custom ICs at 1 Gbps.*

1. Introduction

The introduction of the Asynchronous Transfer Mode (ATM) technology on broadband networks and its capability for combining switching and transport functions, has forced many developers of local area networks to use ATM as the basic technology of their next-generation products [1]. The introduction of cell-based technologies, like Asynchronous Transfer Mode (ATM), in the consumers market, covering the residential and business areas, is based on various transmission media, at different transmission speeds and different topologies. Cell-based systems relay fixed length user cells and establish virtual channel connections before actual data transfer [2].

ITU and the ATM Forum have already defined a set of recommendations which cover various aspects of this technology. These recommendations provide directions for the coding scheme, transmission frame generation, recovery and adaptation, rate decoupling etc. Although the physical layer is mainly based on the Synchronous Optical Network (SONET), other physical layer architectures have been used for providing the basic transmission functions. These architectures are based on existing technologies, which are not acceptable when higher transmission rates are required. Especially in unframed, or pure ATM systems, the functions of the Transmission Convergence (TC) sublayer of the Physical Layer, which are performed at the channel bit rate, must be reconsidered when higher speed is required. The functions of TC are defined independently of the bit timing functions of the Physical Medium Dependent (PMD) sublayer, resulting to implementation difficulties at higher speeds.

Cell-based interfaces are used for forming various types of communications systems but the physical connections between adjacent nodes or between network switches and hosts are based on point-to-point channels. In this paper, a new interface for high-speed cell-based communication systems is presented, emphasizing on its time related functions and their implementability at high data rates. This interface is called VIRUS and its name comes from a special type of idle cells. These idle cells are called Idle Order Sets (IOS), and their acronym IOS is the Greek translation of the word '*virus*'. The VIRUS interface supports point-to-point connections for exchanging ATM cells, based on the services defined by ITU at the physical layer and it is easily implementable at high-speeds. The basic disadvantage of the pure ATM transmission, based either on CMI or NRZ coding and data scrambling, is that the cell delineation mechanism is performed at the bit data rate [3] and that work proposes a new approach for the synchronization problem.

The VIRUS line coding is related with the higher level functions (cell delineation), and is used as the basis for simplifying them. As it will be shown later, most of the physical layer functions are performed at speed lower than the bit data rate at the physical medium. This makes the proposed method capable of supporting high speed communications with low-cost components. The VIRUS interface is defined as a high-speed platform for cell-based intra-office communications, implementing the basic ITU functions of ATM systems and easily connectable to other ATM based networks.

Section II gives a concise description of techniques used in Fibre Channel and ATM for synchronization purposes. The VIRUS interface, its coding scheme and its synchronization functions are described in Section III. Finally, Section IV presents the implementation of the proposed interface for an 1 Gbps ATM Local Area Network (ATM LAN).

2. Coding and Synchronization in Fibre Channel and ATM.

Timing information recovery is performed at various levels of a communication interface and determine various implementation parameters. As the transmission rate increases, the simplicity of the timing mechanisms and their implementability becomes crucial. Timing agreement (otherwise called synchronization) is required at various levels, starting from the clock level and moving up to bit, byte, frame, packet, message etc. [4]. There are various criteria for determining synchronization achievement, like the autocorrelation function of the synchronizing sequence, the Hamming distance etc., but always the principle is to use the shortest possible sync sequence and to minimize the false sync cases. Many synchronization methods have been proposed up to now which handle the bit synchronization problem as independent of the timing recovery at higher layers. A method uses code words with self synchronizing properties, where a combination of patterns with code-violation or non-valid data patterns are transmitted to achieve synchronization. This is the case of the 8B/10B code [5] used in the Fibre Channel [6]. The redundancy introduced by the coding scheme allows the proper selection of the valid code-words for maximizing the Hamming distance. The characters transmitted in the FC are either data or special characters. The combination of special and data characters result in the generation of transmission words which are appropriate for determining character boundaries. The basic advantages of the 8B/10B code are that it is DC balanced, it has near optimum run length and digital sum variation and minimization of encoding time, making it appropriate for high speed transmissions. For optical transmissions, various block codes or bit insertion codes may be used which are bit-sequence independent (BSI), the bit rate increase is kept low and simple circuits can be used [7], [8]. The scrambled NRZ code used in the optical interface of ATM networks does not increase the actual bit rate but also does not satisfy the BSI condition for use at higher transmission rates. The ATM uses also the CMI code (a special type of 1B2B code) for its electrical interface, which has good signal balance but requires twice the information rate as clock frequency.

Some of the above mentioned codes (like the 8B/10B code) provide special characters outside the data alphabet for performing character synchronization, frame delimiting, supplementing the error detection mechanism etc. The ATM performs the functions of byte synchronization, cell delineation, rate decoupling of adjacent nodes and error protection of the cell switching information, using various mechanisms which are independent of its line code. These mechanisms include correlation of the header bits with its CRC, use of idle cells, single-bit error correction etc.

For detecting the boundaries of frames, packets or cells and keeping synchronization, various mechanisms have been proposed [1]. In cell-based networks, cell delineation mechanisms are used, since fixed length packets are transmitted. Many cell delineation mechanisms have been proposed, but the most interesting methods are the use of empty cells, the use of periodic cells as framing patterns and the use of the HEC code [1]. The cell delineation mechanisms follow a three state diagram, consisting of the HUNT, PRESYNC and SYNC states. The HUNT state is used for making initial cell boundary recognition and the PRESYNC state validates this decision by confirming its correctness a number of times. In the SYNC state, the system is synchronized and a number of consecutive validation errors must be detected for losing the synchronization. The *empty cells delineation* method requires the use of a specific pattern in the cell header and boundaries confirmation is performed cell by cell by checking the existence of this pattern. The method's disadvantage is that, under heavy load, it takes some time to acquire synchronization. The *periodic cell delineation* method is based on the enforcement of periodic cells in the user cell stream and thus reduces the available channel bandwidth. Its synchronization (desynchronization) time depends on the timing of periodic cell insertion. The use of the *HEC cell delineation* method, which is used in ATM, is independent of the line coding and of any special patterns but has the disadvantage that malicious simulations of HEC in the cells' payload will result to incorrect synchronization.

The VIRUS interface was developed for supporting high-speed point-to-point communications in an ATM LAN environment, which must be easily interconnected to other ATM networks. So, the ATM functionality was decided to be used but at higher transmission rates, and that resulted to the design of this new interface. The VIRUS physical layer has to provide to the ATM layer the services determined by the ITU recommendations, to operate at higher cell rates and to be easily implementable. The maximum transmission rate in each interface is 1 Gbps and the nodes are interconnected using multimode fibre links.

3. The VIRUS Interface.

When high transmission rates are used, the implementation of the physical layer functions becomes difficult and costly since they are performed over the serial bit stream. The processing speed can be decreased by recovering byte timing using a framing structure, but the use of frames increases the interface complexity. In the VIRUS interface, pseudo-frames are used, which decrease the required processing power, without increasing the

hardware complexity. A pseudo-frame is defined the information transmitted between two control groups of bytes, including the first control group. Before describing the VIRUS pseudo-frames, its coding scheme is presented.

3.1 Line Coding

Since the system requirements were the support of the basic ATM functions at higher speeds than available User-Network Interfaces (UNIs) [3], various coding schemes and transmission systems were considered. The Fibre Channel (FC) technology was the most promising to developing the physical layer. The disadvantage of using the 8B/10B code of FC in a cell-based environment, where single bit error correction is required, is that due to its coding structure, two unacceptable conditions may be met. First, an error bit in the transmitted serial stream may be spread into two bits in the decoded data, making the single error correction capability of TC sublayer non-applicable. Second, a code violation may be obtained erroneously, due to an error occurred in a previous bit position, which altered the Running Disparity of the bit stream, but did not result to a detectable error at the respective character [6]. For these reasons, in the VIRUS interface a combination of two block codes is used. The user data are encoded using the 4B1C code [8], while the 8B/10B code is used only for control and synchronization purposes. The 4B1C code, a complimentary bit is inserted every four bits, is applied twice in each data byte, one time for each nibble, while the idle cells, which are used at the physical layer for cell rate decoupling and cell delineation, are constructed as a sequence of K28.5 and D21.5 (FC terminology) bytes. Following this coding scheme, the Fibre Channel requirement that the maximum run-length of the encoded data does not exceed five bits is satisfied and the correct phase relationship of the extracted clock with the received data is maintained.

ATM uses idle cells for rate decoupling between adjacent nodes. The ATM idle cells are 53 bytes long and their length is equal to the length of the user cells. For rate decoupling purposes, the VIRUS interface uses a smaller group of bytes, which is called Idle Order Set (IOS). Each IOS is composed of two identical transmission words, each transmission word consisting of four characters, two K28.5 special characters and two D21.5 data characters. K28.5 is the Fibre Channel special character used as the first character in each control transmission word. The K28.5 characters are used by the VIRUS receivers for detecting byte boundaries before acquiring cell boundaries. Since the 4B1C coding is used for the user data, the K28.5 character contains code violation and, before achieving synchronization, it can also be detected erroneously at the receiver side in the incoming serial bit stream, as a result of a combination of the two consecutive data bytes, as it is shown in Table 1. In order to minimize this false sync case, two consequent K28.5 characters have been used in the IOS structure. The D21.5 characters have been selected to conform with the FC idle words structure and are used mainly for maintaining the clock synchronization. These two D21.5 characters are not used in the cell delineation procedure and other patterns can be used to support Operation, Administration and Maintenance (OAM) functions. As it will be shown in the synchronization method description, the IOS length must be equal or greater than the cell header length which is five bytes long. For that reason, the IOS was selected to be eight characters long. The Idle Order Sets are used also for the cell rate decoupling mechanism. They are generated at the transmitter side and are removed from the receiver in order to absorb the differences of the actual clocks from their nominal frequencies.

3.2 The Pseudo-frame Structure

The VIRUS physical layer is based on the characteristics of a point-to-point link interconnecting two devices which communicate using the Fibre Channel functions and the ATM method of cells transmission, with the modifications described above. The physical layer of each node is subdivided into two sublayers. The lower sublayer, which is

IOS Structure: **<K28.5> <K28.5> <D21.5> <D21.5> <K28.5> <K28.5> <D21.5> <D21.5>**

Transmission Character	K28.5	K28.5	D21.5	D21.5
Binary equivalent	0011111010	0011111010	1010101010	1010101010

Combinations of valid data words which result to a K28.5 character

XX 00 **1** 1 1 1 10	10XX **X** XXXXX**X**
XXXX **X**XX00 **1**	1 1 1 1 **0** XXXX**X**

Table 1. Notation conventions and IOS structure

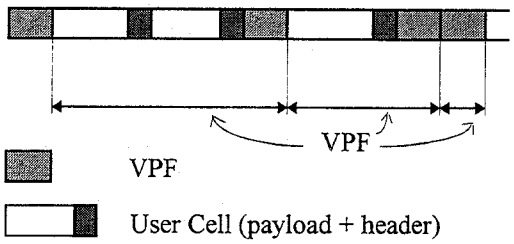

Fig. 1. The pseudo-framing of the VIRUS interface

called VIRUS Channel Adapter (VCA), is related to the functionality of Fibre Channel and deals with physical medium aspects. The upper sublayer, which is called Transmission Convergence (TC) sublayer, deals with the cell stream multiplexing/demultiplexing, cell header error detection and correction, IOS insertion and extraction and coding/decoding of data.

The lower part of the physical layer is based on the Fibre Channel architecture and uses the FC-0 functions of the Fibre Channel (transmission media adaptation, transmitters, receivers and their interfaces) and a combination of the 8B/10B and 4B1C transmission codes. The physical layer uses some of the FC-1 functions (bit and transmission-word boundaries synchronization) and also contains the required functionality for adapting ATM cells to Fibre Channel Datagrams or otherwise called VIRUS pseudo-frames (VPF). As it was mentioned previously, as a VIRUS pseudo-frame is considered the information (number of user cells) contained between two IOSs including the first IOS. Fig. 1 shows an example of VIRUS transmissions and the respective pseudo-frames. The minimum length of a pseudo-frame is the length of an IOS, since the next IOS starts a new pseudo-frame, while the maximum length depends on the rate decoupling mechanism, which determines the maximum number of user cells between two IOSs. The VIRUS pseudo-frames are reconstructed when they pass through a network node and new frame delimiters are generated, while the contained data (cell bursts) are modified.

The VIRUS receiver discards the incoming IOSs and delivers the correctly received user cells to the ATM layer. At the transmit side, the VIRUS accepts user cells from the ATM layer and calculates the fifth byte (Header Error Check - HEC) of the cell header. Then the cell is passed through the VIRUS interface which produces the appropriate VIRUS pseudo-frame format and multiplexes the ATM cells in its structure. The IOSs are generated in the Transmission Convergence sublayer when there are no cells available for transmission or when the number of user cells transmitted after the last IOS has reached a predetermined value. Fig. 2 shows the VIRUS interface state diagram for the transmit direction. It includes a T_X-state counter, c_H,

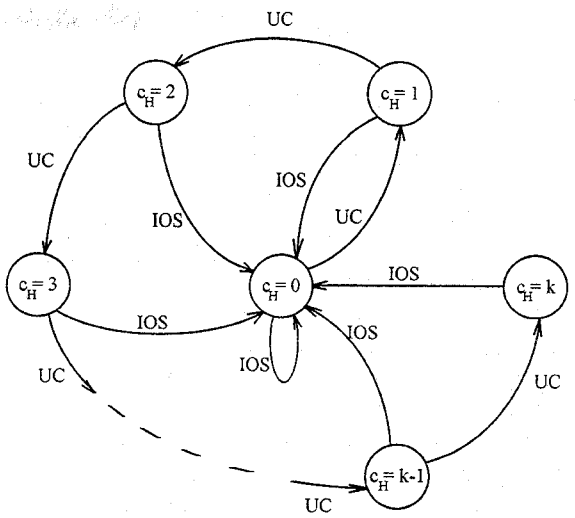

Fig. 2. The VIRUS Tx state diagram

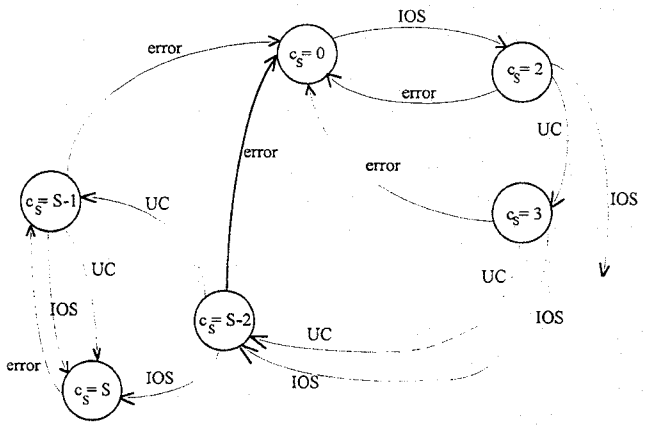

Fig. 3 The Rx-state counter operation.

which indicates the number of user cells transmitted after the last IOS. Initially the c_H is equal to zero. When a user cell is transmitted, the counter increases. When it reaches its maximum value k, the user cell buffer service is ceased, an IOS is transmitted and the T_X-state counter is cleared.

3.3 The Cell Delineation Algorithm

At the receiving side, the IOSs are detected and removed at the TC sublayer where the cell delineation algorithm is implemented. This algorithm is based on the 'HEC' method and on an 'Indication of IOS'. A byte boundaries detector is used which operates under the control of the cell delineation mechanism. The boundaries detector accepts encoded serial data for performing clock recovery and is

capable of detecting K28.5 characters for timing resynchronization when this pattern occurs in the serial bit stream. Whenever the resynchronization part of the boundaries detector is active and a K28.5 character has been detected, an indication is provided. This indication is used by the synchronizer of the TC sublayer for performing the synchronization algorithm. When character synchronization is achieved, the boundaries detector is deactivated for avoiding interpretation of specific patterns in the user data. When resynchronization is required, the character boundaries detector is activated.

Initially the cell delineation mechanism is in the HUNT state and the resynchronization part of the boundaries detector is activated for detecting two consecutive K28.5 characters. Whenever a valid IOS has been detected, the synchronizer changes to the PRESYNC state, the characters' boundaries are considered detected and the synchronizer validates both the cell headers and the IOS structure. The synchronizer uses the Rx-state counter (c_S) which initially is equal to zero and is increased by one or two, depending on the detection of a correct user cell header or a valid IOS respectively (Fig. 3). Since the VIRUS interface uses different lengths for the user cell and the IOS, two validation modules are triggered in each cell or IOS boundary: the HEC module and the IOS module. The HEC module confirms the existence of a cell by validating its fifth byte using the CRC method, while the IOS module matches the first eight characters with the IOS structure. If a valid indication is available at the end of the eighth character, a new cell or IOS boundary is determined. Then, the synchronizer waits for this new boundary to restart the boundary validation procedure. If the synchronizer is in the PRESYNC state, the internal counter is increased once or twice depending on the source of the valid indication. A user cell results to one time increase, while an IOS signals for two times increase, since it contains two FC transmission words. Figure 4a gives timing details of the synchronization method for the different cases of boundary estimation. If no valid indication is available at the end of the eighth character, the validation procedure is restarted. If no valid indication is given for a second time, the validation procedure is cancelled, the internal counter is cleared, the character boundaries detector is activated and the cell delineation mechanism goes to the HUNT state. When cell or IOS boundaries have been established and the internal Rx-state counter gets its maximum value, the synchronizer switches to the SYNC state. These validation procedures are performed in parallel by using the character stream clock, so high processing rate can be achieved, especially compared with the pure ATM cell delineation procedure which is performed on the serial bit stream.

In the SYNC state, the Synchronization State Machine validates the cell boundaries, following a two-step procedure. In each valid boundary, the previously described procedure is performed for estimating the new boundary. If no valid indication is received at the end of the eighth character of this validation, the validation procedure restarts immediately, the internal counter is decreased but the system remains in the SYNC state. At the same time, a new cell or IOS boundary is estimated, by adding 53 character times to the previous boundary. A cell header is corrupted with higher probability than an IOS, since it uses more bits (50 line bits instead of 20). If the cell header or IOS validation procedures do not give a positive result for a second time (Fig. 4b), the Rx-state counter is cleared, the cell delineation mechanism goes to the HUNT state and the character boundaries detector is activated.

4. The VIRUS Interface Implementation.

The VIRUS Interface has been implemented using commercially available components and a custom chip-set, prototyped using FPGAs. The VIRUS Channel Adapter (VCA), which implements the physical medium dependent functions, is composed of a GaAs TAXI chip-set and optical interface devices while TC uses a custom chip-set. Fig. 5 shows the VIRUS interface block diagram. The interface with the ATM layer is performed using 32-bit wide busses for decreasing its operational speed and different paths are used for the cell header and the cell payload. The data rate in the channel is 1 Gbaud/sec, which results to 800 Mbit/sec for the user data, due to the 8B/10B and 4B1C coding schemes.

The data at the two busses are processed independently and are multiplexed with the output of the IOS Generator at the last multiplexing stage. When the transmission of the fortieth byte of the payload of a cell begins, the TC transmitter checks the availability of a new cell in the ATM layer. If no cell is available the IOS Generator is activated. In the case where a new cell is available, the first four bytes of the cell header are transferred and stored in the pipeline module, while the HEC byte is calculated and the 4B1C coding is performed. When the payload transmission of the previous cell has been finished, the header of the next cell is stored in the Physical layer pipeline structure and is ready for transmission. Whenever a user cell follows an IOS, its header is processed during the IOS transmission. Although the operational need of the circuit is 100MHz, this parallel architecture allows the CRC generation to be performed at 25MHz. In order to achieve the total of 100 Mbytes/sec system operation, the 32-bit data bus is multiplexed in three stages and the 4B1C coding circuit has been implemented in four different transfer paths. The Tx TAXI components are used to serialize the data and feed the optical transmitter. The VCA module recovers the data clock and transfers the encoded data to the TC Receiver. It also provides an indication if a correct K28.5 pattern has been detected in the incoming data stream.

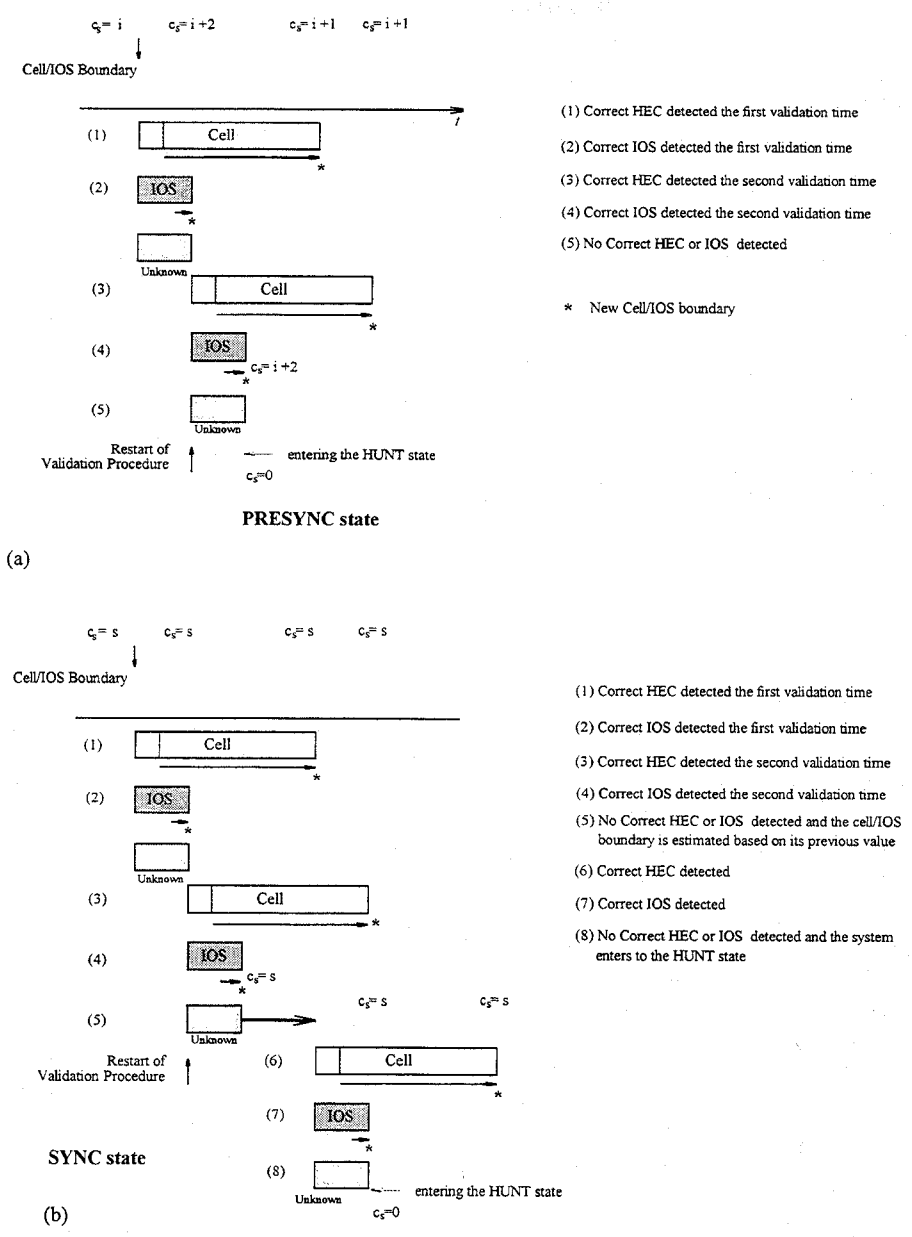

Fig. 4 Cell/IOS boundary validation procedure

The PRESYNC indication along with the output of the IOS detector and the HEC Validation and Correction signal are the required information for performing the previously mentioned synchronization method in the Synchronization State Machine. The first four bytes of the received header are decoded and stored in the User Cell Header Buffering for eight byte cycles. The payload bytes received during this period are also stored in the Cell Payload Buffer and the HEC validation and correction function is performed. If more than a single error have been detected in the received header, the cell is discarded, otherwise the position (byte number and bit position) of the error is calculated and the header correction is performed during the transfer of the header to the ATM layer using a XOR-gate plane [9]. The TC transmitter has been implemented in a XC3164 FPGA, while the TC Receiver has been implemented in two XC3164 FPGAs due to its complexity.

The TAXI transmitter is a commercially available GaAs IC, which receives a 10-bit parallel bus and generates the NRZ bit stream for driving the optical transmitter. The TAXI receiver accepts the NRZ serial data from the optical receiver, recovers the data clock by using its internal clock recovery PLL, re-establishes byte boundaries, whenever is required, using the K28.5 pattern and generates a 10-bit

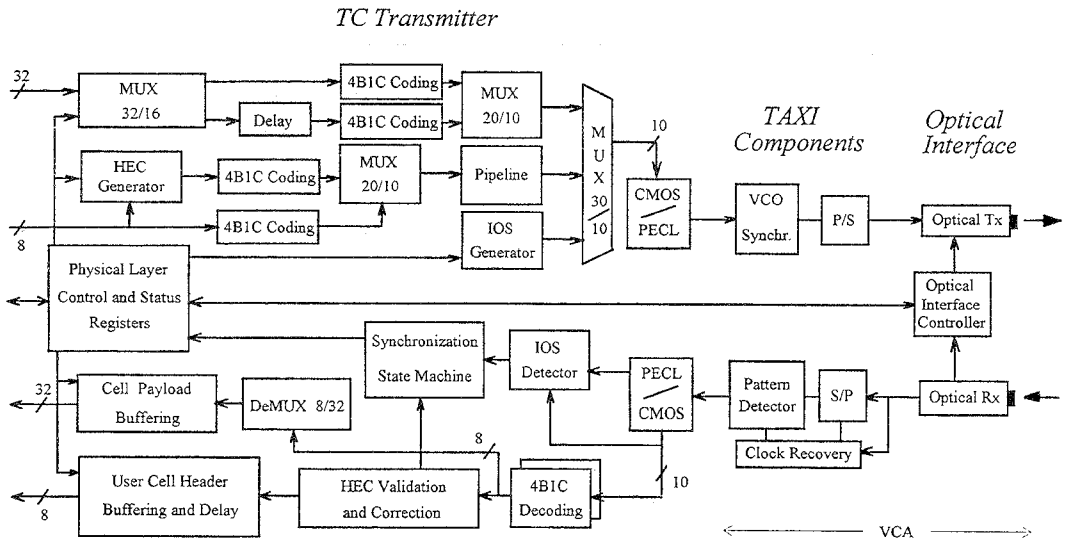

Fig. 5. The VIRUS interface block diagram.

synchronous parallel bus for further processing by the TC sublayer.

For the implementation of the optical interface, the FTM-8500 (Tx) and FRM-8500 (Rx) data link modules of FINISAR Corp. have been used. These modules use short wavelength lasers to transmit data at 1 Gb/s rate through multi-mode fibre (62.5/125). These two optical modules are controlled by the FCC-2000 controller achieving full control and performance monitoring of the upstream and downstream optical links. At the transmitting side the optical power launched into the fiber is controlled while the received optical power at the photodiode is measured in combination with the module temperature. These facilities allow the node management unit to determine the status of its upstream and downstream links, allowing the higher layers' management to take the appropriate corrective actions during a failure in the optical link.

5. Conclusions

In this paper, a new interface called VIRUS for use in cell-based systems was presented. The interface supports the implementation of various functions (cell delineation, cell header error detection and correction, synchronisation etc.) for transmitting a cell stream using the basic functions of the Fibre Channel. Although is uses a combination of block codes, single bit error correction is performed and various operations are executed using the same hardware modules. The main advantage of the proposed interface is its implementability in high speed links. In the paper, the architecture of a prototype at 1 Gbps was presented and was indicated how a high cell transmission rate of 1.88 Mcells/sec can be achieved using commercially available FPGAs to implement the functions of the VIRUS interface.

References

[1] S. Walters, D. S. Burpee and G. H. Dobrowski: "Evolution of Fiber Access Systems to ATM Broadband Networking", *Proceedings of the IEEE,* Vol. 81, No. 11, November 1993, pp. 1588-1593.

[2] Martin de Prycker: "Asynchronous Transfer Mode: Solution for Broadband ISDN", Ellis Horwood Ltd, 1991.

[3] ITU Recommendation I.432: "B-ISDN User-Network Interface - Physical Layer specifications", June 1992.

[4] D. Wiggert: "Code for Error Control and Synchronization", Artech House, 1988.

[5] P. A. Franaszek and A. X. Widmer: "Byte Oriented DC Balanced (0,4) 8B/10B Partitioned Block Transmission Code", *USA Patent,* No. 4,486,739, December 4, 1984.

[6] "Fibre Channel - Physical and Signaling Interface (FC-PH) Rev. 3.0", ANSI X3T9 I/O Interface, Working Draft, June 1992.

[7] N. Yoshikai, K. Katagiri and T. Ito: "mB1C Code and Its Performance in an Optical Communication System", *IEEE Transactions on Communications,* Vol. Com-32, No. 2, February 1984, pp. 163-168.

[8] Y. Takasaki, M. Tanaka, N. Maeda, K. Yamashita and K. Nagano: "Line Coding for Very High Speed LANs", *Electronics Letters,* Vol. 26, No. 24, November 1990, pp. 2051-2053.

[9] A. Maniatopoulos, T. Antonakopoulos and V. Makios, "A New Single - Bit Error Correction Circuit for ATM Interfaces", *Electronics Letters,* IEE, Vol. 31, No 8, April 1995, pp. 617-618.