An Implementation of the 25.6Mbps ATM Physical Layer Interface

Chan Kim, Hun Kang

Electronics and Telecommunications Research Institute

Abstract

In this paper, the design and implementation of 25.6Mbps ATM physical layer interface is described. This interface has data rate of 25.6Mbps and uses UTP cable. It adopts scrambling and uses 4B5B coding for data randomness and cell delineation. The specification was implemented for a B-NT2 system called CANS and can connect subscribers up to 100ms apart using UTP-3 cable. Finally some design considerations are addressed.

1. INTRODUCTION

There have been proposed various types of physical layer interfaces for ATM and among them are 155Mbps STS-3C, 100Mbps TAXI, 25.6Mbps UTP etc. These interfaces have been implemented by many vendors and are now being used by many users. Of these interfaces, 25.6Mbps UTP is expected to be the most popular interface for desktop application in the future. Currently the most widely used network is Ethernet in which many users share the 10Mbps bandwidth. The 25.6Mbps interface is relatively easy to implement and cost effective while providing "sufficient" bandwidth for now. This interface is similar to 100Mbps TAXI[1] but to reduce electro-magnetic interference problem, scrambling technique is used.

2. FEATURES OF THE SPECIFICATION

The interface has logical bit rate of 25.6Mbps, which means it operates on 32Mbaud since it adopts 4B5B coding. And this interface specification requires the transmitter and receiver to be able to send and receive data through upto 100ms of UTP-3 with BER < 10E-10. In this section, the transmission convergence layer of the specification is explained.

2.1 4B5B Block Coding and Decoding

The 4B5B Block coding provides the means for cell delineation and scrambler/descrambler reset. It also supports periodic timing signal for isochronous services. It provides over 3 transitions per 5 bit symbol and run length is limited to less than or equal to 5. Of the 32 possible symbols, 17 symbols are valid in this implementation. The commands are composed of the Escape(X) followed by any symbol of which commands only X_X and X_4 and X_8 are defined. The cell delineation is accomplished by prepending X_X or X_4 to each ATM cell before transmission.

X_X : Start-of-cell (with scrambler reset)

X_4 : Start-of-cell (without scrambler reset)

The X_8 is used to transmit synchronization character for isochronous services(like 8kHz or 2.43Mhz for SRTS application). Figure 1 shows the format of cell transmitted on the line.

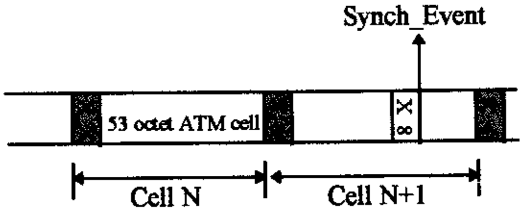


Figure. 1 Example of Cell Delineation and Sync_Event using Commands

2.2 Cell Scrambling and Descrambling

To provide an appropriate frequency distribution of the electrical signal across the line, the data are scrambled before transmission. The scrambler and the descrambler are each comprised of a 10-bit PRNG. The PRNG is based on the following polynomial;

$x^{10} + x^7+1$

In the specification, the serial version of the PRNG is shown but for implementation purposes, byte operating version was devised. The (de)scrambler must be either clocked or preset at a given clock period. During idle states, arbitrary data continue to be encoded and scrambled to maintain synchronization of the receiver PLL.

2.3 NRZI Encoding and Decoding

In order to bound the run length of 1s or 0s during transmission, data symbols from the encoder are serialized and NRZI coded before transmission. This makes the differential signaling efficient.

2.4 HEC Generation and Verification

The HEC covers the entire cell header. The transmitter calculates the HEC value for the first four octets of the cell header, and inserts the result into the HEC field. It is the remainder of the polynomial x^8 multiplied by the content of the header excluding HEC field. The pattern 01010101 is XORed before being inserted. Cells with incorrect HEC are discarded at the receiver.

The low clock rate makes it easy to implement this interface and using UTP cable eliminates the need for expensive optical transmitter and receiver.

3. HARDWARE IMPLEMENTATION

This interface has been designed and implemented in ETRI for a B-NT2 system called CANS(Centralized Access Node System). The board is one of the subscriber interface board along with 155Mbps STM-1 and 100Mbps TAXI interface boards.

Figure 2 shows the hardware block diagram of the interface.

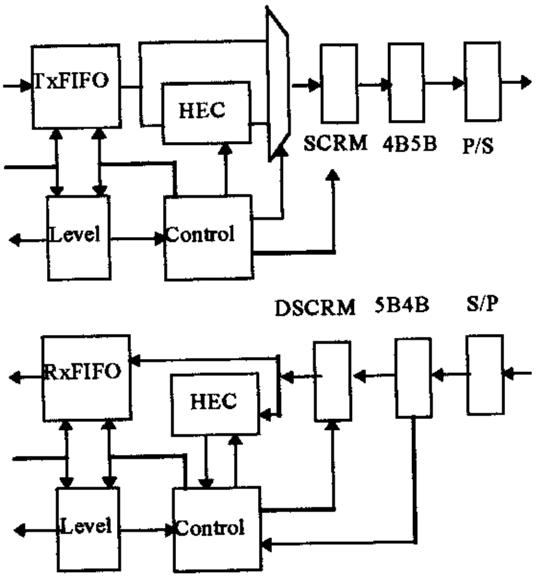
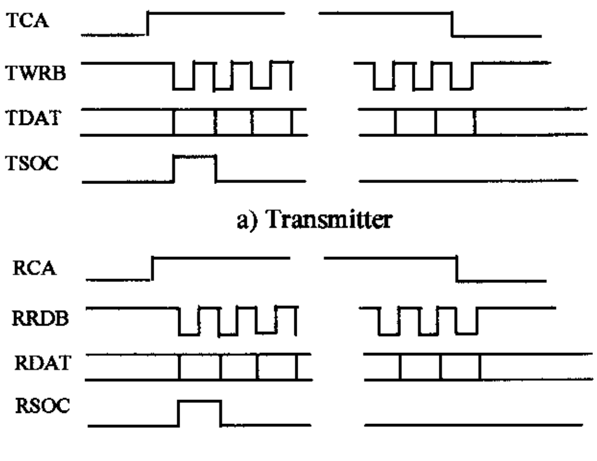


Figure. 2 Hardware Block Diagram

The interface board has TX-FIFO and RX-FIFO and two A1240A FPGAs to implement transmitter and receiver logic respectively. Two level counters were implemented for transmitter and receiver each to initiate some routines or to inform the ATM layer circuit of cell or buffer space availability. Figure 3 shows the interface timing diagram between ATM layer and physical layer.



b) Receiver

Figure 3. Interface Timing Diagram between ATM Layer

3.1 Transmitter Implementation

The transmitter consists of FIFO buffers, a buffer level counter, HEC generator, scrambler, 4B5B encoder, parallel to serial converter and controller. The level counter keeps the byte level of the FIFO using the write(TWRB) signal and the read signal of the FIFO. When the level is greater than 53 after the ATM layer writes one or more cells into the buffer, the transmitter controller starts its cell transmission routine. It reads 53 bytes from the buffer and inserts HEC in the fifth byte of the data. The data are scrambled, 4B5B encoded and shifted out serially onto the transceiver with serial clock. The transceiver converts the signal into differential analog signal and sends it to magnetic module. The signal is launched to the physical copper cable through transformer.

The transmitter controller's state diagram is not shown here because it is enough to explain in words. The controller waits in idle state and as soon as the GT53 signal from the level counter becomes true, it starts its cell transmission sequence. The state machine moves through five header states and one payload state. On entering the payload state, it loads a down counter which counts from 47 down to 0. When the counter reaches its final value 0, the statemachine gets out of payload state. If there is another cell left in the TX-buffer after finishing a cell transmission, the controller jumps to an appropriate state so that no clock cycle is lost during contiguous cell transmission. FIFO read signal and appropriate control signals for HEC encoder, scrambler,mux are generated. The counter is disabled in idle state.

3.2 Receiver Implementation

The receiver consists of transceiver which performs clock and data recovery with equalization, serial to parallel converter, 4B5B decoder, descrambler, HEC verifier, receiver FIFO buffer, a buffer level counter and controller. The signal from the UTP cable is received by transformer and recovered by the transceiver and sent to the S/P converter with recovered serial clock. The byte clock generator, by resetting its internal counter, adjusts the clock phase every time a start-of-cell is detected. The deserializer converts the data into byte stream using aligned byte clock. The data are 4B5B decoded and then sent to the descrambler. When X_X is received , the descrambler is preset and when X_4 was received, it is not preset but clocked. The descrambled data are then shifted through 4 registers until HEC verification is finished. When HEC is verified to be correct, the cell data are written into the FIFO from the first octet of the cell. The level counter counts the buffers level using the write and read(RRDB) signals of the FIFO and generates RCA signal which indicates to the ATM layer that a cell is waiting in the rx-buffer. The receiver does not check the rx-buffer space availability during a cell reception because the ATM layer is considered to be fast enough to process the cells. In case of receiver buffer full, an interrupt is generated so that CPU can reset the receiver buffer.

The receiver controller's state diagram can also be explained briefly. The controller waits in idle state and when X_X or X_4 signal is detected from the 4B5B decoder, it moves through five header states and one payload state. On entering the payload state, it loads a down counter which counts from 47 down to 0. When the counter reaches its final value 0, the statemachine comes back to idle state. The FIFO write is delayed 5 bytes from all header and payload states. When a new start-of-cell is detected during a cell reception, the former cell is discarded just 1 clock before the start of the new cell write cycle. FIFO write signal and appropriate control signals for HEC decoder, descrambler are generated with state transitions.

4. DESIGN CONSIDERATIONS

There are a few considerations to be made.

4.1 FIFO level counter design

There are several ways of implementing FIFO level counter. We could use an up-down counter arbitrating the read/write signals to count up or down since the up-down counting cannot be performed in both direction at the same time. This method requires fast logic to sample the read/write signals and determine the direction and is not recommended in high-speed design. We can also use an up-counter and a down-counter to count in each direction and add the results to get the level. This is the method used in this design. Another good approach is to use a cell counter in which the level is counted by cells(of course, two 53 byte counters are needed in each direction).

4.2 Initializing receiver controller

In designing the receiver controller, if we want the controller to go to a known state after power-up, the jump condition should be just idle pattern since we cannot expect the idle pattern to come up immediately after power-up. This makes the controller jump to idle state whenever the idle pattern is received.

4.3 Noise problem

In analog chip, the noise in the power pin directly and adversely affects the signal hence deteriorating the chip's performance. Especially when the receiver is equipped with an equalizer, special care should be taken for the receiver circuit to prevent the strong transmitter signal from getting into the receiver by interference. If not appropriately decoupled, this transmitted signal gets into the receiver signal path and gets recovered by the receiver. This makes it difficult to implement a signal detect logic. To avoid this irritating problem, the transmitter and receiver should be as far apart as possible and the differential signals must go parallel and close to each other. In the analog part, the receiver's VCC and GND plane should be separated from that of the transmitter. Through experiment we had some difficulties in the analog part because the recever was so sensitive(it has equalizer to adapts to the received signal condition) that the transmitted idle pattern signal is received in the receiver when the line is disconnected at the other end of the cable.

4.4 Clearing FIFO during the operation

Due to its nature, FIFO devices cause problem if one or more garbage data(not whole sized data like one cell) happen to remain in it. If this happens for any reason, later reading of the data becomes obsolete and further processing is impossible(The hardware runs but with wrong data!)

Since the ATM layer board is allowed to send cells only when it has been signaled by the physical layer that the transmitter buffer has space for at least one cell, the TX-FIFO never goes into full state in normal operation. To prevent the ATM layer from causing garbage in the FIFO, the TCA and RCA signal should be made low during FIFO reset thus preventing ATM layer from starting to write or read during FIFO reset and thus preventing the possibility of writing or reading across reset's final edge. To prevent the physical layer circuit from making garbage data, new reading or writing of cell data should also be disabled during FIFO reset condition, so that when reset signal goes high, the circuit cannot be in the midst of reading or writing cell data. In addition to these protection mechanisms, the software FIFO reset signal should be guaranteed to be long enough to let any on-going writing or reading operation finish.

By this means, the FIFOs can be made to be in empty state and garbage free after FIFO reset. The FIFO reset is done when the system is reset and when the board is put into a slot on the fly, and when the FIFO has been detected to have gone full. And since the FIFO-clear is the ORed version of reset and S/W FIFO clear command, this FIFO protection applies to H/W, S/W reset and S/W FIFO clear. By this means, after FIFO-clear is finished, the circuit starts over from empty FIFO.

5. CONCLUSION

In this paper, an implementation of 25.6Mbps ATM physical layer interface is described. This interface has data rate of 25.6Mbps and uses UTP cable. It adopts scrambling, 4B5B and NRZI coding for data randomness and cell delineation. The specification was implemented for a B-NT2 system called CANS in ETRI and through experiment it was verified that cells can be transmitted without error to over 100m's of UTP3. Finally some design considerations were addressed. Due to its low clock rate, the digital circuit can be easily implemented without much difficulties and the use of UTP cable makes it possible to use low-cost transceiver and magnetic module. This shows that the specification's main purpose of low cost implementation can easily be achieved and 25.6Mbps ATM is the strong candidate for future widly used ATM network interface.

REFERENCES

1. The ATM Forum, "Physical Layer for 100 Mbps Multimode Fiber Interface", ATM User-Network Interface Specification Version 3.0, September 1993.

2. The ATM Forum, "ATM Physical Medium Interface Specification for 25.6Mb/s over Twisted Pair Cable", ATM_Forum/94-1008r0.1, November 1994.

3. Chan Kim and Jung Hoon Paik, "An Implementation of 100Mbps TAXI Subscriber Board", 20th Conference on LCN(to be published), October 1995.