IMPLEMENTING A PREAMBLE OA&M CHANNEL IN A 10 GIGABIT PER SECOND ETHERNET IC

by

David Heath Culley
BASc., Simon Fraser University, 1996

PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING

In the
School of Engineering Science

SIMON FRASER UNIVERSITY

Spring 2007

APPROVAL

Name: **David Heath Culley**

Degree: **Master of Engineering**

Title of Project: **Implementing a preamble OA&M channel in a 10 gigabit per second ethernet IC**

Supervisory Committee:

Chair: **Dr. Shawn Stapleton**
Professor
School of Engineering Science, SFU

Dr. Daniel Lee
Senior Supervisor
Associate Professor
School of Engineering Science, SFU

Dr. Stephen Hardy
Supervisor
Professor
School of Engineering Science, SFU

Stephen Dabecki
External Examiner
Director – Product Development
PMC-Sierra, Inc.

Date Defended/Approved: April 10, 2007

ABSTRACT

For this project, I led a team to implement a proposed OAM (Operation, Administration & Maintenance) protocol layer for a 10GE Physical Layer IC (Integrated Circuit). This report gives background of the protocols and technologies involved, and the history of the OAM proposal. It also describes my work to implement the proposal.

The proposed OAM protocol makes use of the eight-byte preamble on every Ethernet packet which is not needed for full duplex Ethernet links. Although the IEEE 802.3 working group ultimately rejected this protocol, it is a technically superior approach, and was interesting to implement.

Keywords:
10GE, 10 Gigabit Ethernet, OAM

Subject Terms:

I dedicate this work to my adorable wife who has never stopped encouraging me to improve myself. Without her patience and hard work with the kids, I would not have been able to complete this.

I love you, Liliane!!

ACKNOWLEDGEMENTS

I would like to thank the team of talented designers from PMC-Sierra who worked with me on this project. Steve Dabecki, as my manager, was an excellent guide and mentor in this project. Mohamed Allam and Alison Xu worked on the detailed design and implementation of the circuit blocks described in this work. Gary Bourque, as a participant in IEEE served as an excellent mentor of all things Ethernet.

I also want to thank the professors on my supervisory committee and the graduate secretary for their support and willingness to work with some unexpected distractions I faced in preparing this work. Thanks for your help and support, Dr Lee, Dr Hardy and Raj.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

10GE	Ten Gigabit (10,000,000,000 bits) per second Ethernet
8B10B	**8-B**it to **10-B**it encoding – An encoding of 8-bit words into 10-bit code-space used by XAUI.
ASSP	**A**pplication **S**pecific **S**tandard **P**roduct – An IC designed for a specialized function and an open customer base.
ATM	**A**synchronous **T**ransfer **M**ode – A layer 2 protocol using a steady stream of fixed size cells instead of bursty packets.
CPE	**C**ustomer **P**remises **E**quipment
CRC	**C**yclic **R**edundancy **C**heck – A mathematical operation that generates data check-sums based on a serial feedback equation.
CSMA/CD	**C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **D**etection – The official title of "Ethernet" in the 802.3 specification.
DFT	**D**esign **F**or **T**est – Circuitry is added to an IC to allow for manufacturing testability.
EPON	**E**thernet **P**assive **O**ptical **N**etwork – An access technology using optical fibre to the home, where passive components allow multiple transmitters on a single fibre.
Ethernet	A networking technology defined by the IEEE 802.3 CSMA/CD specification.
FCS	**F**rame **C**heck **S**equence – The last field of an Ethernet packet containing a check-sum of the data in the packet.
FIFO	**F**irst **I**n **F**irst **O**ut – A data structure (often a RAM or block of registers) where data is read out in the same order as it is written in.

Gbps	Gigabit per second
GMII	**G**igabit **M**edia **I**ndependent **I**nterface – An interface specified in 802.3 between the MAC and PCS layer for Gigabit Ethernet.
HDLC	**H**igh level **D**ata **L**ink **C**ontrol – A byte based packet protocol for generic communications.
HEC	**H**eader **E**rror **C**heck – A CRC check-sum of an ATM cell's four byte header
IC	**I**ntegrated **C**ircuit (a semi-conductor chip)
IPG	**I**nter-**P**acket **G**ap – The space between Ethernet packets. IPG must be at least 12 bytes, and has no upper bound.
LAN	**L**ocal **A**rea **N**etwork (typically covering an office or work-group)
LSB	**L**east **S**ignificant **B**it – The bit in a field or byte with the lowest mathematical value.
MAC	**M**edia **A**ccess **C**ontrol – Ethernet protocol layer responsible for packet addressing and filtering.
MSB	**M**ost **S**ignificant **B**it – The bit in a field or byte with the highest mathematical value.
OA&M	**O**peration, **A**dministration and **M**aintenance – Overhead functions related to network status, link monitoring, etc.
PCS	**P**hysical **C**oding **S**ublayer – The Ethernet protocol layer responsible for line coding and data serialization.
PHY	**Phy**sical Layer device (Layer 1 on the OSI model)
PLL	**P**hase **L**ocked **L**oop – A circuit where an oscillator tracks the phase transitions of an input signal.
PMA	**P**hysical **M**edium **A**ttachment – Attaches the PCS layer to the PMD layer

PMD	**P**hysical **M**edium **D**ependent – The Ethernet protocol layer responsible for driving and detecting signals on a given physical medium (such as a wire or optical fibre).
RS	**R**econciliation **S**ublayer – Ethernet protocol layer between the PCS and the MAC.
SFD	**S**tart **F**rame **D**elimiter – The last byte of an Ethernet preamble before the Destination Address field. Has a value of 10101011.
SONET	**S**ynchronous **O**ptical **NET**work – A synchronous frame based transport protocol defined by Bellcore GR253 specification. This is used by telephone providers and other long distance, high reliability networks.
VHDL	A **H**ardware **D**escription **L**anguage used to define digital functionality of an IC.
VLAN	**V**irtual **LAN** – A method of creating multiple logical networks within a broader Local Area Network.
WAN	**W**ide **A**rea **N**etwork (typically spanning large distances, and making use of external service providers)
XAUI	10 Gigabit **A**ttachment **U**nit **I**nterface – An interface specification for 10Gbps data over 4 serial links operating at 3.125 Gbps each (to accommodate for 25% coding overhead of 8B10B).
XGMII	10 **G**igabit **M**edia **I**ndependent **I**nterface – An interface specified in 802.3 between the MAC and PCS layer for 10 Gigabit Ethernet.
XSBI	10 Gigabit **S**ixteen **B**it **I**nterface – A competing PMA interface to XAUI.

1. INTRODUCTION

In 2002, I lead a development team of engineers to design and bring to production a 10 Gbps Ethernet PHY ASSP (Application Specific Integrated Circuit) for XAUI based optics. In the course of this project, I supervised and was ultimately responsible for all aspects of chip design including: functional specification, team management, design capture in VHDL, functional verification, DFT (Design for Test) implementation, physical layout and timing closure. Since my employer is a fabless semiconductor supplier, an outside supplier did the prototype fabrication, but when the prototypes came back from the foundry, as the project leader, I was involved in their DFT testing, electrical characterization, functional validation, product engineering and customer documentation.

The chip my team developed was part of a family of 10 Gbps PHY chips, and we re-used some of the main systems in our chip from the earlier projects. However, I was heavily involved in specifying and implementing the RS and OAM layers of the 10GE PHY device. For this report, I will first give some background on the Ethernet protocol and application that this device was targeted for in chapter 2. Chapter 3 introduces the OAM protocol proposal that was built into this chip, and chapter 4 goes into my own implementation of that protocol and the RS sub-layer. Chapter 5 concludes with mention of the final state of this project, and some recommendations for future work.

2. BACKGROUND

2.1. Ethernet primer

Many people recognize Ethernet as a ubiquitous LAN (Local Area Network) technology that connects the computers in an office, workgroup or home. It is perceived as cheap, simple to implement, and easy to maintain, and has by far become the dominant network protocol in the LAN.

Ethernet is defined by the 802.3 working group of the IEEE, and its details are specified in the 802.3 specification. Over the years, the 802.3 specification has expanded from a simple and cheap CSMA/CD network on a shared medium (a wire) to a whole host of data rates, transmission mediums, and associated protocols. Various sub-groups of the IEEE meet every two months in cities all over the world to discuss and ratify amendments to the 802.3 specification so that Ethernet will keep pace with advances in technology.

Protocol stack

Before one can begin to understand what is going on with a device used in an Ethernet system, the Ethernet protocol stack must first be introduced.

Figure 1 OSI Protocol Stack

Data unit	Layer	Function
Data	Application	User application (eg. Netscape)
	Presentation	Data format & encryption (eg. HTTP)
	Session	Inter-host communication (eg. FTP)
Segments	Transport	End-to-end reliability (eg. TCP)
Packets	Network	Logical addressing (eg, IP)
Frames	Data Link	Physical addressing
Bits	Physical	Media, line coding, electrical levels, physical connectors, etc

Ethernet sits at layers 1 and 2 of the OSI Protocol Stack in the Data Link and Physical layers (See Figure 1). 802.3 further defines several sub-layers that go together to implement the Ethernet stack.

Figure 2 Ethernet Protocol Stack

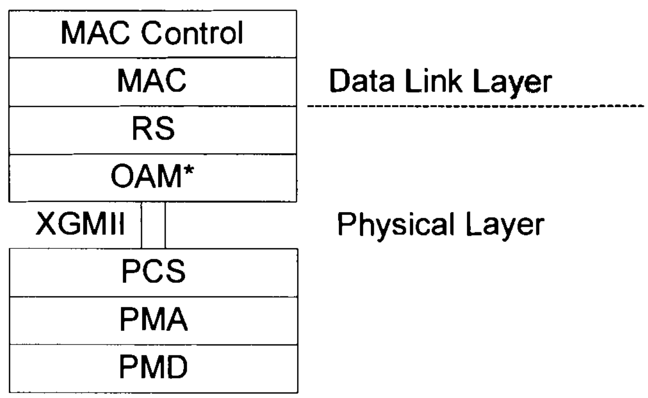


Figure 2 roughly represents the general case of the Ethernet protocol stack, but here I explain the terms as they relate to 10GBASE-X, which is the Ethernet variant implemented in my chip. The * in the OAM layer of Figure 2 indicates a proposed layer that is not officially part of the 802.3 specification.

The MAC Control layer sits above the MAC layer, and is beyond the scope of the chip I worked on, and of this project.

The MAC (Media Access Control) layer handles such things as physical address filtering, protocol error filtering, and statistics gathering (such as number of packets dropped, size of packets passed, etc). The MAC is implemented in the chip I worked on, but as this functional block was re-used from previous chips without any modification, I will not go into any details on its implementation.

The RS (Reconciliation Sublayer) layer adapts the bit-serial protocol of the MAC layer to the parallel encodings used by the PHY. For the case of a XAUI PCS, the RS layer would map the bytes from the MAC into 8B10B codes. The OAM (Operation, Administration and Maintenance) layer was proposed to sit under the RS layer for the protocol that was implemented in this project. The 802.3ah working group voted down this proposal, and it is not currently here in the 802.3 specification (see section 3.1). Instead, the current specification has the OAM layer as an optional layer above the MAC Control layer. As the blocks I was deeply involved in for this chip sit at the RS and proposed OAM layers, I will be going into more detail on these below.

XGMII (10 Gigabit Media Independent Interface) is a 32bit full duplex logical interface specification. The 802.3 specification states that "it need not be implemented, but its specification provides a common reference point for the definition of the layers above and below". For other data rates, similar interfaces are specified such as GMII for Gigabit Ethernet, and MII for 100Mbps and 10Mbps Ethernet.

The PCS (Physical Coding Sublayer) is different for each physical medium implementation. For a XAUI Physical interface, the PCS encodes the four 8-bit words on the XGMII interface into four 8B10B code words for transmission/reception on the physical medium. 8B10B maps the 256 values of 8-bit words into the 1024 code-space of 10-bit values. The 10-bit code words are selected such that they either contain an equal number of 1's and 0's, or else are paired in complements (where the same 8-bit word maps to two 10-bit words – one with more 1's, and the other with more 0's). A running count is kept of the relative number of 1's and 0's, and when an imbalance is detected, the complement value is chosen that will bring the count back into balance. The result is that when the 10-bit code-words are serialized, a minimum transition density is guaranteed (no more than five 1's or 0's in a row). The guaranteed transition density ensures that clock and data recovery on the receiver won't drift and cause bit errors. The 10-bit encoding also guarantees that the number of 1's transmitted will equal the number of 0's so that an electrical medium would not incur a DC bias. Un-utilized points in the 8B10B code-space (because we are encoding only 256 values) allow control words such as <SOP> (Start of Packet), <EOP> (End of Packet) and various IPG control words (used to align the four lanes) to be encoded in-band.

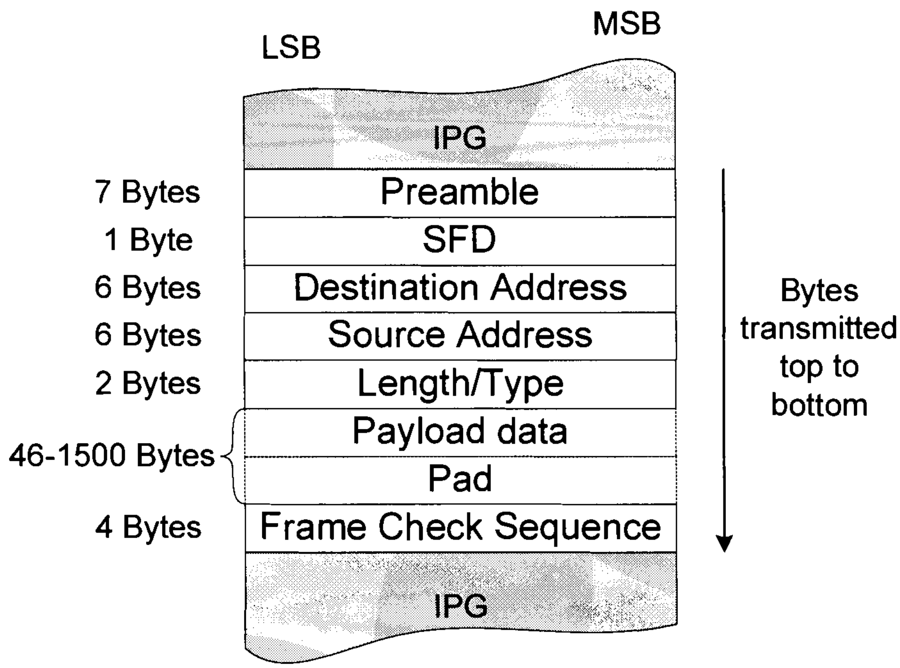
The PMA (Physical Medium Attachment) layer serializes the 10-bit code words from the PCS to send to the PMD (Physical Medium Dependant) in the transmit direction. In the receive direction, the PMA recovers the clock (by means of a PLL) and deserializes the code words from the PMD to the PCS. The PMA is also responsible for data alignment. In the case of XAUI, the data is transmitted to and from the PMD as four serial streams, which could have varying latencies (typically a few nanoseconds of fixed latency due to board trace mismatches). In this case, the PMA must guarantee that the four streams are aligned as one group of four code-words.

The PMD layer is implemented in an off-chip optics module and is beyond the scope of this report. The PMD would be concerned with physical line coding, and wavelength and intensity of the lasers used to communicate over the fibre optics.

Packet format

802.3 defines a packet based network protocol where packets can be anywhere from 64 bytes to 1518 bytes (VLAN tags add 4 bytes to these sizes). In practice, some networks allow "jumbo packets" up to 9600 or more bytes. Ethernet packets are byte based with the LSB (bit 0) transmitted first, and they follow the structure shown in Figure 3. Packet size is calculated from the Destination Address through the Frame Check Sequence. Note that the packet format applies to all variants of Ethernet (so some fields may seem redundant in a given implementation).

Figure 3 Ethernet Packet Format



IPG (Inter-Packet Gap) is the gap between packets. An Ethernet packet is only sent when one is available. At all other times the line is occupied by IPG. At the MAC, the IPG is required to be minimum 12 bytes, with an unbounded maximum. At the PCS, the IPG may occasionally shrink to 5 bytes due to code word alignment and inter-clock boundaries.

The **Preamble** is seven bytes of 01010101 patterns, and is intended to allow collision detection in legacy half-duplex networks (if two hosts start transmitting simultaneously) and data synchronization at the receiver. The **SFD** (Start Frame Delimiter) has a pattern of 11010101, and indicates the end of the preamble, and that the next byte starts the Ethernet packet. Sometimes in this paper, I refer to an 8-byte preamble. In these cases, it is the Preamble + SFD to which I am referring.

The **Frame Check Sequence** (FCS) is calculated from a CRC (Cyclic Redundancy Check) over all bytes from the Destination Address through the Payload data and Pad inclusive. Its generating polynomial is:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

As the MAC receives a packet, it calculates its own CRC, and if the calculated CRC does not match the FCS, the packet is recognized as errored, and is discarded.

The remainder of the fields are defined in 802.3, section 3.2, and since they are all concerned with the MAC layer, they do not concern my work in this project.

10GE PHY layer

XSBI

Several variants of 10GE have arisen that primarily differ in how they electrically interface to the laser and optical receiver that is the source and sink of signals on the optical fibre. An earlier variant of 10GE is 10GBASE-R, which uses an XSBI (10Gbps Sixteen Bit Interface) PCS implementation. XSBI uses a 64B/66B encoding to communicate with the optical module (a single module that houses the laser, lenses, optical receiver, and some signal conditioning electronics) over a 200 or 300 pin interface. This specification was solidified earlier than XAUI and 10GBASE-X, and was easier to implement reliably, but the chip, board design and optical module are very expensive (which goes against the market perception that Ethernet is cheap).

XAUI PHY

A newer specification, 10GBASE-X uses the four link XAUI protocol to communicate with the optical module. XAUI is defined by 802.3 in chapter 48, and was briefly described in the PCS section of pages 4 and 5. It uses four links of 8B10B encoded data transmitted and received in differential pairs at 3.125 Gbps per link for a 16 pin data interface. Optical modules using a XAUI interface are smaller and cheaper than XSBI based modules. A further advantage of XAUI modules is that they can be pluggable (a great benefit for a high reliability WAN network).

2.2. 10GE WAN

Collision-less network

This project is particularly interested in the 10 Gbps data rate (10GE). A 10GE node is typically not a part of an office LAN, but rather is a conglomeration of many LANs' traffic for switching in a metro core, or for long distance transmission to another city (these network elements would typically form part of the WAN). Of particular note, is that at this data rate, and for any appreciable distance (anything more than a few feet), wires are simply not capable of carrying the signal. Thus, 10GE uses lasers and fibre optics as the physical medium rather than electrical signals on a wire.

An important result of using fibre optics as the physical medium is that it typically[1] cannot be a shared medium. For 10GBASE-X all network connections

---

[1] There are exceptions to this generality, such as EPON, but it has an additional layer of protocols to handle multiple transmitters on a fibre that is terminated at a single endpoint.

are point-to-point, and traffic is routed by a switch, rather than by a target host listening for its own traffic on the medium.

Maintenance channels

Another requirement of a WAN network is that certain overhead is required to maintain the network itself. This overhead is typically needed for querying the status of network elements and links, reporting status and alarms, and for configuring the network to address weak points that are detected.

Earlier data networks had accomplished this requirement by packaging the data protocol (such as Ethernet or ATM) into a frame network that has OAM built in, such as SONET. Adding OAM capability to Ethernet allows the service providers to remove a protocol layer, and simplify their network equipment deployment. They can deploy Ethernet switches alone, and not need SONET Add-Drop Muxes. Thus, it was argued that a carrier class 10GE solution should also have OAM capability.

3. PROTOCOL FOR PREAMBLE OAM

Ethernet is ubiquitous in the Local Area Network, and recent advances in technology such as the advent of GE and 10GE are allowing Ethernet to make inroads in the Metro and Wide Area Networks. In 2001, the 802.3ah working group was formed to bridge these two domains with Ethernet in the First Mile (EFM). The goal of the EFM working group is to add the features to Ethernet that would make it suitable for Service Providers, and thus give rise to a pure packet based network from beginning to end.

3.1. OAM at IEEE specification working group

At the March 2001 meeting of the IEEE 802.3ah working group, Hiroshi Suzuki and Bruce Toley of Cisco Systems, Inc. proposed that an OAM protocol be considered for inclusion into the Ethernet spec (802.3). It was argued that with Ethernet making its way into the metro and access network, service providers would require an OAM capability in order to detect and recover from link faults, measure link performance, and otherwise maintain the network.

The 802.3ah working group voted in favour of adopting an OAM proposal, and work began in earnest toward developing this specification. Over the course of the next 14 months, the proposal for OAM in Ethernet took shape, as Hiroshi Suzuki, and his allies developed their preamble specification, and Denton Gentry of Dominet Systems and his allies developed their competing MAC packet based specification (Getting an engineering specification ratified by a standards body can be very political).

Advantages of the Preamble OAM proposal

Mr Suzuki's proposal is built on the observation that since 10GE exclusively uses a point-to-point network topology and regular data transitions in the IPG that allow continual clock recovery, the 7 byte preamble defined for every Ethernet packet is redundant, and could be taken over for other uses without any impact to functionality. This proposal had 3 important factors in its favour:

- It does not require any extra bandwidth to implement OAM. So data bandwidth in the network is not impacted.
- Because the OAM sublayer exists between the PCS and MAC layer, the OAM protocol and related data is invisible to the MAC layer, and is thus more secure than a packet based OAM.
- OAM data is sent with every packet, so very fast response time to network faults can be assured.

Advantages of the Packet OAM proposal

Mr. Gentry's proposal is built on the simplicity of deploying a packet based protocol over a packet network. It has several clear advantages over the Preamble proposal:

- No new hardware is required to implement the Packet OAM protocol, and the new software is not overly complex.
- The Packet OAM protocol inserts OAM packets at regular intervals, so it does have a small impact on data throughput, but that is still a very small portion of the available bandwidth.

- One of the greatest advantages of the Packet OAM proposal is that it can be applied to any generation of Ethernet – even those that actively use the preamble.

Standards adoption

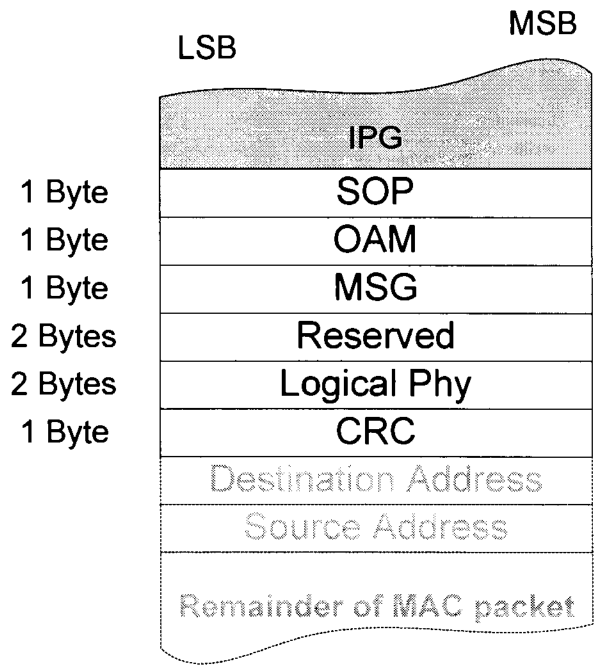
At the May 2002 meeting in Edinburgh, Scotland the 802.3ah working group voted in favour of Mr Gentry's proposal, which was then refined and finally became section 57 of the 802.3 specification. In its final form, the preamble proposal was added as an option to the Packet OAM proposal as a means of insuring quick response time to network faults, but in the end, all forms of preamble OAM were rejected by the working group.

Throughout the 14 month life of the preamble OAM proposal, it continued to change and take shape. The following sections are drawn from the content of some of Mr Suzuki's later slides, and give the details of the proposed protocol.

3.2. Frame format

As indicated in the baseline proposal for Preamble OAM at the March 2002 802.3ah meeting, the preamble OAM takes over the 8 byte preamble+SFD of every Ethernet packet, with the sequence of bytes shown in Figure 4.

Figure 4 OAM Preamble Format

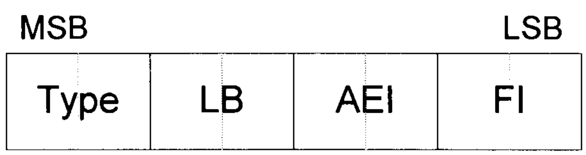


The SOP is not technically a part of the OAM preamble, but is reserved for the PCS layer, since many PCS protocols require a special data word in the SOP byte. The Reserved bytes are not defined for this protocol, but could be used for vendor specific information, or for some unknown future need.

The OAM byte carries message flags in its bits as indicated in Figure 5.

OAM byte

Figure 5 OAM Byte Format



The two bit Type field indicates the type of preamble as shown in Table 1.

Table 1 Preamble Type Field

MSB | LSB	Meaning
0 | 0	OAM Preamble with Data
0 | 1	Standard Preamble
1 | 0	Dummy Frame without Data
1 | 1	Reserved

OAM preamble is indicated by 00 or 10 in the type field. 01 in the type field is part of the 01010101 default preamble value, and indicates that no OAM is present in the preamble.

The Loop Back (LB) field is used to ping the remote end of the link. The local OAM transmitter will send 01 in the LB field as a ping request, and then wait for a timeout. When the remote end receives the 01 code in the LB field, it will respond with 10 in its LB field back to the local host. If the Local host doesn't get the 10 code back before the timeout expires, then it will know that the remote host is down. 00 in the LB field indicates no operation, and 11 is reserved.

The Asynchronous Event Indication (AEI) field is used to alert the far end of an impending failure according to Table 2. As a last gasp, when the host detects one of these failure conditions, it will send the AEI code as long as it can to alert the network of the failure.

Table 2 Asynchronous Event Indication Field

MSB | LSB	Meaning
0 | 0	Normal Operation
0 | 1	Pending failure due to power off
1 | 0	Pending failure due to node fail
1 | 1	Pending failure due to other reasons

The Fault Indication Field (FI) allows the RS layer's Remote Fault and Local Fault indication to be reflected at the OAM layer. If a fault is detected with a remote host, the MSB is set to 1 to indicate the remote fault state. If a fault is detected at the local host, the LSB is set to 1 to indicate a local fault state. At the far end, the OAM layer would read the FI bits, and pass Fault Indications out of band to the RS layer.

The Message Byte (MSG)

The Message byte serves as a "serial link" between local and remote hosts, and is used for both transmit and receive. The protocol was not defined in Mr. Suzuki's slides, but could be something like an HDLC packet protocol, where one byte of the packet is sent with each OAM preamble. This message channel would be used to access remote management parameters, and report network statistics.

Logical PHY

The two byte Logical PHY field is required by EPON networks, where a single fibre may have multiple transmitters on it. The PHY id, allows the OAM packets from one CPE transmitter to be distinguished from the OAM packets of other transmitters.

CRC

The CRC byte was not explicitly defined in the March 2002 presentation by Mr. Suzuki, but it would likely be calculated over the seven preceding bytes of the OAM preamble with a generating function similar to the HEC byte of an ATM cell header ($G(x) = x^8 + x^2 + x + 1$).

3.3. Data rate considerations

Ethernet MAC packets are only sent when there is data to send. In periods between bursts of data, the Ethernet link can be idle for quite a while. This data burstiness would create an unacceptable unbounded minimum data rate for OAM preambles without some further modifications to the Preamble OAM protocol.

Mr Suzuki's March 2002 presentation describes an OAM "Dummy Frame", which is essentially an OAM preamble that is inserted into the IPG in the absence of MAC packets. There are several rules that govern when an OAM Dummy Packet can be inserted. First of all, the 12-byte minimum IPG must be maintained before and after the Dummy Packet. Secondly, a minimum gap between Dummy Frames needs to be defined. This gap would give the minimum OAM data rate in a long IPG. In the example Mr. Suzuki gave, a 76-byte gap between Dummy Frames would correspond to the same OAM data rate in the IPG as would naturally arise from a constant stream of 64 byte packets.

With the above considerations, the bandwidth of the OAM channel is defined with both upper and lower bounds. The maximum OAM data rate occurs in a constant stream of minimum sized MAC packets (or similarly defined Dummy Packet inter-gap in a long IPG). This maximum OAM data rate is:

$$\text{DataRate}_{\text{Max}} = \frac{2 \text{ byte (OAM)}}{8 \text{ byte(preamble)} + 64 \text{ byte (packet)} + 12 \text{ byte (IPG)}} = 2.4\%$$

The minimum corresponds to a steady stream of maximum sized MAC packets (with VLAN tag). The minimum OAM data rate is:

$$\text{DataRate}_{\text{Min}} = \frac{2 \text{ byte (OAM)}}{8 \text{ byte(preamble)} + 1522 \text{ byte (packet)} + 12 \text{ byte (IPG)}} = 0.13\%$$

For 10GE, these bounds correspond to a minimum OAM bit rate of 1.29 Mbps, and a maximum OAM bit rate of 23.8 Mbps. Note that these bit rates are only for the defined OAM and Message byte. Bandwidth can be increased by making use of the Reserved bits in the OAM preamble. Note that even the minimum data rate is orders of magnitude greater than the bandwidth that would be required for voice communication, so it should be sufficient for any OAM traffic requirements, which would typically require much less bandwidth.

3.4. Interfacing to the MAC Layer

In order for the OAM layer to interface correctly with legacy RS and MAC layers that expect a standard preamble, it may be necessary in the receive direction for the OAM layer to write the OAM preamble back to a normal preamble. To do this, the first seven bytes of the preamble (SOP through Logical PHY) should be written to 01010101 as for a normal preamble, and the last byte (CRC byte) should be written to 11010101 as the SFD.

4. MY WORK ON OAM PROTOCOL

When I was given the task of implementing the preamble based OAM protocol in silicon, the proposal was still developing in the standards body, and had not taken its final form, nor been rejected. Implementing a network protocol in silicon provides for a higher processing rate than any software implementation, but it has the drawback that the circuits tend to be hard-coded and very expensive to change (a simple chip revision can easily cost over a million dollars). I had to design an OAM layer processor (as described in Chapter 3) that would have the inherent efficiencies of a dedicated hardware implementation, but also the flexibility to accommodate a final specification that could differ from the understanding we had at the beginning of the project. So the greatest challenge in designing the OAM layer for my ASSP was to build in flexibility. To do this, I defined the OAM functions with many operating modes which are selectable by software settings.

4.1. Fixed assumptions of the OAM implementation

The implementation of OAM layer had to make several assumptions as a skeleton around which the flexible implementation could be built. Firstly, any preamble which could potentially contain an OAM header must have 7 bytes of preamble plus SFD. The 802.3 standard specifies that 8 such bytes precede each packet, but collisions and other line effects can cause more or less bytes to be received. This assumption is fairly safe for the ASSP that I was working on, because 10GBASE-X does not allow collisions, and the IPG line coding does not look like the preamble (it is conceivable that some PCS layers could have alternating 1/0 IPG – but the XAUI implementation does not).

A further assumption that is required by the PCS layer below the OAM layer is that the first byte of the 8 byte preamble cannot be used for OAM data. The first byte of the preamble is not used because the PCS layer inserts a special SOP codeword into the first byte, and would overwrite any data there.

The last byte is not allowed for user configurable data because it is reserved for a CRC byte to ensure data integrity of the OAM preamble. An error check byte is required in the preamble header, since nothing else within the protocol stack can detect and ensure data integrity on the preamble header.

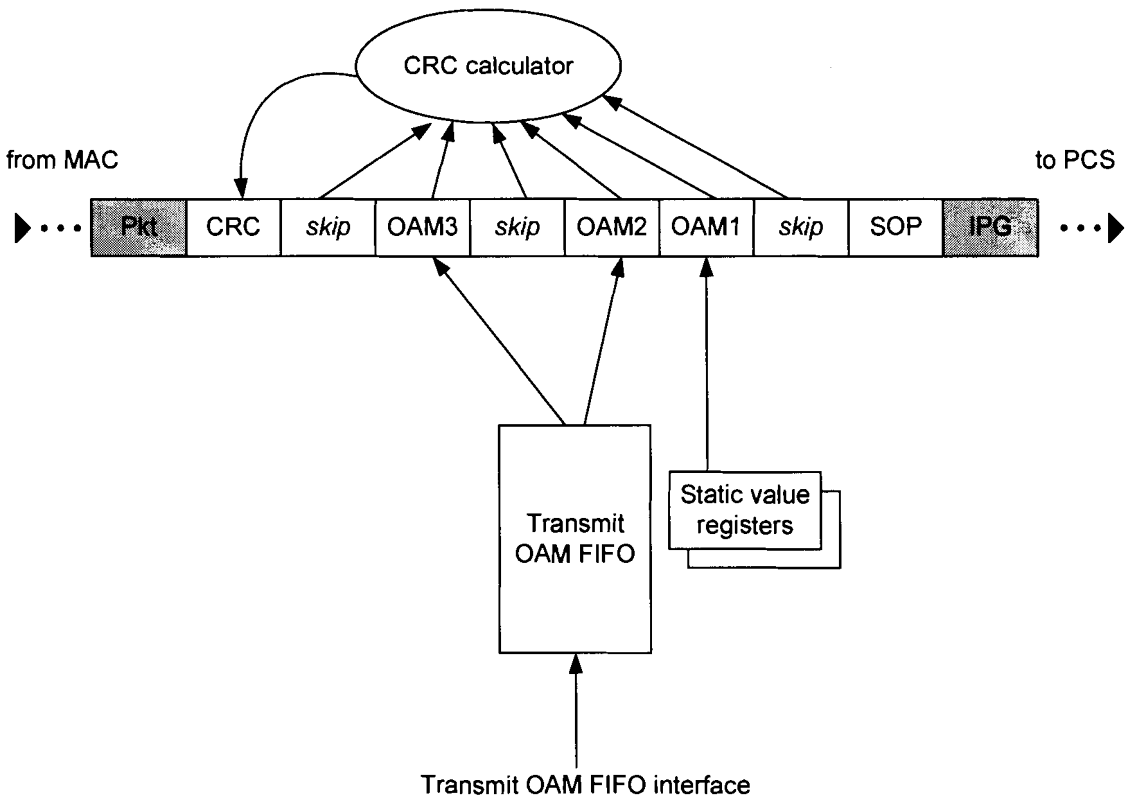
4.2. Flexible byte types

To maximize flexibility of the OAM implementation, we provide the capability to insert and extract up to six bytes from the preamble of each packet through a transmit and receive FIFO interface. In this manner, an off-chip FPGA or network processor can monitor and configure the bytes that go into the preamble, and our ASSP prototype can be manufactured before the OAM specification is complete.

Transmit OAM

The transmit OAM processor can be configured to overwrite any of the 6 available preamble bytes with OAM bytes, or to pass the MAC generated preamble bytes through transparently. Each of the bytes to be over-written can optionally be sourced from a static micro-processor configured register, or from the transmit OAM FIFO interface. Figure 6 shows an example where static value registers overwrite preamble byte 3, the OAM FIFO overwrites bytes 4 and 6, with bytes 1, 5 and 7 passed through transparently. In this example the CRC is calculated over the 6 preamble bytes after SOP.

Figure 6 Transmit OAM implementation



Static registers can be used to implement fixed values for reserved bytes, or slow changing or static values such as identification codes or status flags. The FIFO interface can be used for per-packet status flags or a packetized communication channel.

An error check sequence is calculated over a configurable set of preamble bytes using a CRC-8 Header Error Check (HEC) with the generating polynomial

$(G(x) = x^8 + x^2 + x + 1)$. This HEC can have a programmable 8-bit value XORed with it, and the result is optionally placed in the last preamble byte (overwriting the SFD). The XOR mask allows the ASSP to compensate for any static bytes that should have or should not have been included in the CRC calculation.

Receive OAM

The receive OAM processor allows any of the six preamble bytes between the SOP and the SFD/CRC to be extracted to the receive OAM FIFO interface.

Up to two bytes of the preamble can be used as filtering bytes to determine which packet preambles contain OAM data to be further processed. The selected filter bytes are compared against a programmable filter mask, and the result is used to optionally pass or drop the OAM preamble bytes to the OAM FIFO interface. This filtering can be used if not every preamble contains OAM data, or if an OAM preamble is received that is destined for a different device ID.

A CRC-8 Header Error Check is calculated over selected bytes of the received preamble using a generating polynomial $(G(x) = x^8 + x^2 + x + 1)$, and the result is compared against the final byte of the preamble. The Receive OAM processor can be configured to drop the OAM data from that preamble if the calculated and received OAM headers do not match.

Figure 7 Receive OAM implementation

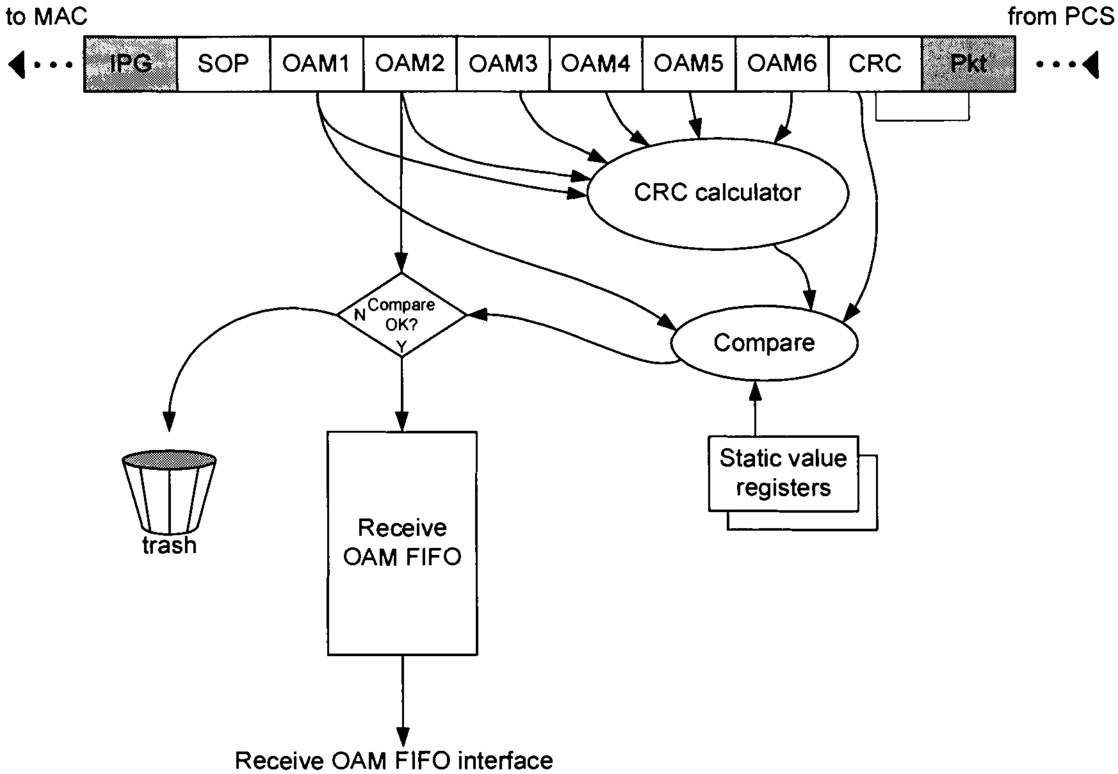


Figure 7 shows a receive OAM implementation where the value from byte 2 of the OAM preamble is stored in the OAM FIFO if the CRC over the whole preamble matches the CRC in the OAM header, and if the value in OAM1 register matches the value in the static value registers.

The Receive OAM processor can optionally write the preamble bytes back to 01010101, and the HEC byte to 11010101 so it looks like a normal preamble with SFD to a downstream MAC processor.

4.3. Mini-packets

The Transmit OAM processor creates the Dummy Packets described in Section 3.3, but with some modifications. Since we could not be sure what kind of data density would be required, we implemented a configurable gap between the previous OAM preamble and the insertion of a Dummy Packet. This gap-before parameter can be used to set the maximum data rate OAM preambles in continuous IPG. There is also a user configurable gap-after parameter that can be used to ensure that the minimum IPG is maintained after a Dummy Packet. No Dummy Packet would be generated if the gap between successive Ethernet preambles is less than <gap-before> + 8 bytes + <gap-after>. In addition to these, the minimum IPG of 12 bytes between the EOP of the preceding packet is guaranteed before any Dummy Packet is inserted.

4.4. Blank preambles

Another issue that needed to be considered in the absence of a finalized specification is what to do when an Ethernet packet is passing the transmit interface, but there is no OAM data to insert. To solve this problem, we again made the behaviour configurable. The Transmit OAM processor can either insert a pre-programmed static value into the preamble, or pass the preamble untouched.

5. CONCLUSION

5.1. Current state of project

At the time of this writing, the ASSP that I worked on to implement the preamble based OAM is in production, and is being used in customer systems. As mentioned above, the preamble based OAM proposal did not become part of the 802.3 standard, but some customers still make use of the features we inserted for it in their products. Customer confidentiality restricts me from saying more about how these features are being used, but it is sufficient to say that the effort was not wasted.

Since this ASSP entered the production phase of its life cycle, many changes have continued to occur in the Ethernet specification, and new flavours of 10GE are available. As mentioned, new protocols are continually added to the specification, and semiconductor companies and system vendors continually take risks to implement expected features that may or may not finally be ratified into the specification. Those that take the risk could win, and be the first to the market with a new protocol, or they could lose and have wasted their effort to add a feature that would never be used. Those that take their time have the luxury of implementing a fixed function for a solid specification, and don't have to waste logic gates on the flexibility that is required early in a protocol's life cycle. However, the late-comers may give up market share, which could also be a costly risk.

5.2. Recommendations for future work

There are several directions that future work on this subject can take. The protocol that was implemented in this project allows a proprietary network to "hide" a secure communication channel on the links between network nodes. This channel was intended to be used as a standardized OAM channel, but many other uses could be found for this channel. Networks built with equipment from the same vendor could run a proprietary OAM protocol over the preamble data channel to achieve the quick response time that is not possible with the standardized MAC packet based OAM protocol. Such networks could have a competitive advantage over more standardized networks in the marketplace.

Another possible future use for the preamble data channel could be to guarantee a quality of service for a particular class of data without affecting the random traffic that is otherwise present in the network by effectively creating a hidden network within the OAM preamble. This could be useful for such applications as Voice or Video over IP, where latency of the network is important.

If future research were to look into testing out these or similar network applications, the fact that the silicon already exists in the ASSP we created would make the research that much easier.

REFERENCE LIST

IEEE Computer Society, Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, IEEE 802.3-2005, 9 December 2005

PMC-Sierra, Inc; PM3393 S/UNI 1x10GE-XP Single Chip 10 Gigabit Ethernet LAN PHY for XAUI-Based Optics Short Form Data Sheet; PMC-2012355; April 2002

PMC-Sierra, Inc.; S/UNI 1x10GE-XP ASSP Telecom Standard Product Data Sheet; PMC-2020726; December 2006 (password required)

PMC-Sierra, Inc.; Receive OAM Injector for 10Gbps Ethernet LAN TSB Engineering Document; PMC-2020006; Feb 2003 (PMC internal document)

PMC-Sierra, Inc.; Transmit OAM Injector for 10Gbps Ethernet LAN TSB Engineering Document; PMC-2020007; Apr 2003 (PMC internal document)

Gentry, Denton, Dominet Systems; OAM in Frames, presentation slides for March 2002 IEEE 802.3ah EFM Task Force; http://grouper.ieee.org/groups/802/3/efm/public/mar02/gentry_1_0302.pdf

Squire, Matt, et al.; EFM OAM Baseline Transport Proposal, presentation slides for May 2002 IEEE 802.3ah EFM Task Force; http://grouper.ieee.org/groups/802/3/efm/public/may02/squire_2_0502.pdf

Squire, Matt, Hatteras Networks; EFM OAM Work Plan, meeting minutes slides for May 2002 IEEE 802.3ah EFM Task Force; http://grouper.ieee.org/groups/802/3/efm/public/may02/squire_1_0502.pdf

Suzuki, Hiroshi, Cisco Systems; OAM on Preamble Baseline Proposal, presentation slides for March 2002 IEEE 802.3ah EFM Task Force; http://grouper.ieee.org/groups/802/3/efm/public/mar02/suzuki_2_0302.pdf

Suzuki, Hiroshi & Tolley, Bruce, Cisco Systems; Why OAM for Ethernet, presentation slides for March 2001 IEEE 802.3ah EFM Study Group; http://grouper.ieee.org/groups/802/3/efm/public/mar01/suzuki_1_0301.pdf

Suzuki, Hiroshi, Cisco Systems; OAM in Preamble FAQ, presentation slides for May 2002 IEEE 802.3ah AFM Task Force; http://grouper.ieee.org/groups/802/3/efm/public/may02/suzuki_1_0502.pdf

Suzuki, Hiroshi, Cisco Systems; PHY OAM Baseline Proposal, presentation slides for July 2002 IEEE 802.3ah EFM Task Force; http://grouper.ieee.org/groups/802/3/efm/public/jul02/oam/suzuki_oam_1_0702.pdf