IEEE 802.3
Higher Speed Study Group

XGMII Proposal

9/26/99
York, England

Ken Barker, John Ewen, Dennis Karst, Jeff Lynch, Dono Van-Mierop, Fabrice Verplanken, Al Widmer	**IBM**
Paul Bottorff, Norival Figueira	**Nortel Networks**

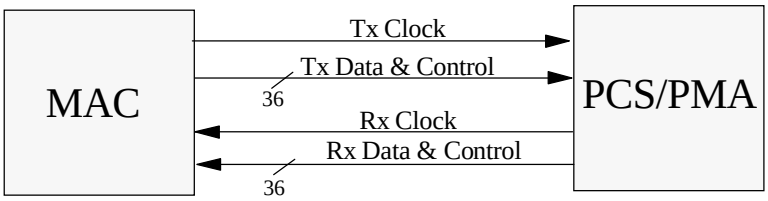
Objectives

for the

XGMII Interface between the MAC and PHY

- **Build on Frazier, et al. MAC/PCS Parallel Interface Proposal to use embedded delimiters, rather than discrete signals**
- **Increase MAC/PLS Interface data rate to 10.0 Gbps**
- **Support 10 Gbps or slower PHYs**
- **Provide efficient data handling**

Parallel Interface Proposed by Frazier, et al. (July, 1999)



- **32 data bits, 4 control bits (one per byte), one clock, for transmit**
- **32 data bits, 4 control bits (one per byte), one clock, for receive**
- **Dual Data Rate (DDR) signaling, with data and control driven and sampled on both rising edge and falling edge of clock**
- **Control bit per byte allows use of embedded delimiters, rather than discrete signals**
- **Control bit per byte allows interface to be scaled in speed and width**

Parallel Interface Control (Frazier, et al.)

- **Control bit (C) is "1" for delimiter and special characters**
- **Control bit (C) is "0" for normal data characters**
- **Delimiter and special character set includes:**
  - **IDLE**
    - during the inter-packet gap
    - when there is no data to send
  - **SOP**
    - for one byte duration
    - at the beginning of each packet
  - **EOP**
    - for one byte duration
    - at the end of each packet
  - **ERROR**
    - when an error is detected in the received signal
    - when an error needs to be forced into the transmit signal
- **Delimiters and special characters are distinguished by the value of the 8 bit data bundle when the corresponding control bit is "1"**

Increase MAC/PLS Data Rate to 10 Gbps

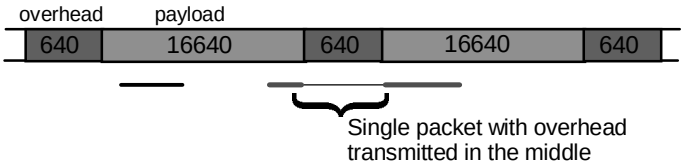
- **As suggested by various proposals, increase MAC/PLS interface data rate to 10 Gbps by:**
  - **Using a 4-byte (word) wide data path**
  - **Clocking data on both edges of the clock**
  - **Using a 156.25 MHz clock**

Support 10Gb/s or Slower PHYs

- **MAC/PLS Interface runs at 10.0 Gbps**
- **To accommodate slower PHYs such as OC-192:**
  - **Adopt a word (4-byte) based pacing mechanism**
  - **Add TX_WH ("word hold") from PHY to MAC**
  - **Add NULL special control character**
    - To specify invalid bytes within a frame transfer
    - Always inserted in groups of 4 (full word)
  - **Provides simple synchronization**
    - MAC withholds one transfer cycle of data in response to each TX_WH
    - A full word of NULL within 0 to "N" transfer times of TX_WH going active
    - Value "N" to be defined as upper bound by standard
- **Word-based flow control minimizes FIFO size in PHY**
  - Size is bounded and deterministic
  - FIFO size is independent of link speed up to 10 Gbps

Alternative Proposals

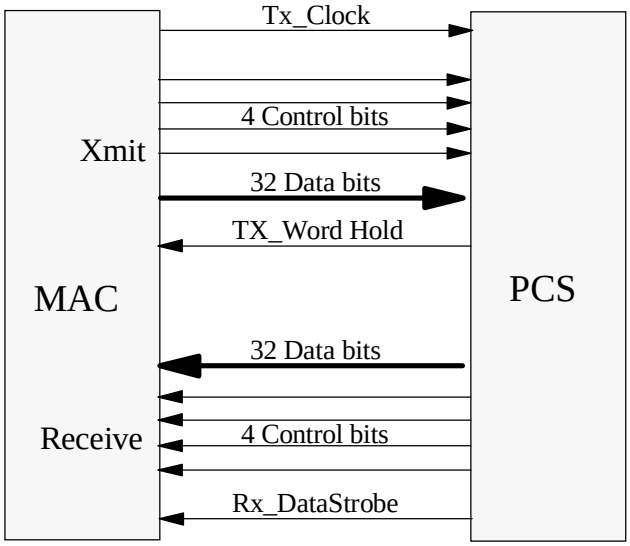
- **Multiple IPG stretching schemes have been suggested**
  - **With or without using a hold signal from the PHY**
  - **Incorporated into the MAC or in a shim between MAC and MAC control**
- **Since these schemes operate between frames, they require relatively large PHY buffers**
- **In the case of OC-192c:**
  - **Transmit path: > 640 bytes of buffering**
  - **Receive path: > 640 bytes of buffering**
    - Up to the maximum frame size of 1522 bytes



Word-by-Word Hold Is the Best Solution

- **No PHY buffer is required to insert overheads**
  - **FIFO buffer only to handle XGMII transfers**
- **No PHY buffer is required to receive packets crossing overhead**
- **Supports any encoding system: 8B/10B, length type, byte stuffing**

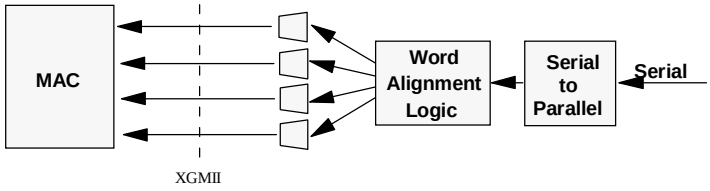
XGMII Interface



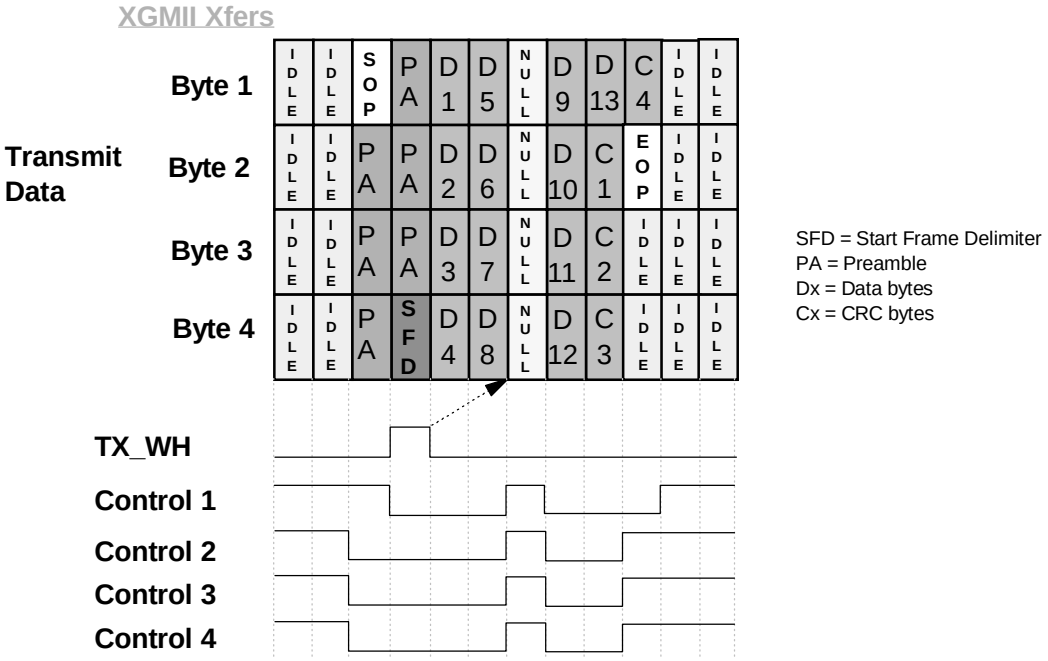
- **Change the name of Receive Clock to Receive DataStrobe to indicate that it is derived from the Tx_Clock (wrapped back)**

Provide Efficient Data Handling

- **Efficient data handling at 10 Gbps is essential**
- **To ensure efficient processing of data by the MAC, both transmit and receive frames should be aligned on word boundaries**
  - Likely that data will be stored in word wide (or wider) memory
  - Eliminates realignment of receive data to word boundaries by MAC
  - **MAC transmits all frames aligned on word boundaries**
  - **Incoming frames are word aligned by PHY for transfer across the XGMII**
    - Frame transfer starts after a SFD is detected
    - PHY already performs byte to word manipulation



Transmit Example



Further Efficiencies?

- **Carry word-based coding thoughout transmission scheme (XGMII & PCS)**
- **Make SOP, SFD, Preamble and IPG symbols 4 bytes long (i.e., one word)**
  - **Treat EOP as a special case**
    - Use multiple encodings to ensure word alignment
    - Similar to the technique used by 802.3z (T, TR)
- **Require that any frame manipulation (such as IPG shrinkage) always occur in 4-byte chunks (1 word)**
  - **Granularity of a single 4-byte symbol**
- **This allows PHY and MAC both to operate on word boundaries for efficient data handling**

Summary

- MAC/PLS Interface data rate should be 10.0 Gbps
- XGMII should be 32 bits wide
- XGMII Pacing Mechanism should be word-oriented
- Frame transfers across the XGMII should start on word boundaries
  - Both transmit & receive frames
- Further efficiencies can be gained by extending word based alignment and coding throughout the transmission scheme