Firmware Design Document

Module Name: transmit engine

Author: Paul
Project Leader: Zheng Cao
Date: 5th November 2005

This page is left intentionally blank.

Revision History

Date	Author	Issue	Comment
05/11/2005	P.L.	1.0	First Version

Reference

[1] 10G Ethernet Mac System Design Issue 1.0
[2] Xilinx LogiCORE 10-Gigabit Ethernet MAC User Guide
[3] IEEE 802.3ae Media Access Control (MAC) Parameters, Physical Layers, and Management Parameters for 10 Gb/s Operation

Contents

List of Tables

List of Figures

1 Introduction

The document describes the design of the transmit engine used in the Opencores 10 Gb Ethernet project. 10 Gb Ethernet is part of the IEEE 802.3 standard. It is essentially a faster version of the Ethernet where half duplex operation mode is not supported.

The MAC design is loosely based on the Xilinx LogiCORE 10-Gigabit Ethernet MAC, where the transmitter and the receiver incorporate the reconciliation layer. Therefore the transmit engine will be specifically designed to interface the client and the physical layer.

2 Detailed Design

2.1 Module Description

The transmit engine provides the interface between the client and physical layer. Figure 2-1 shows a block diagram of the transmit engine with the interfaces to the client, physical, management and the flow control.

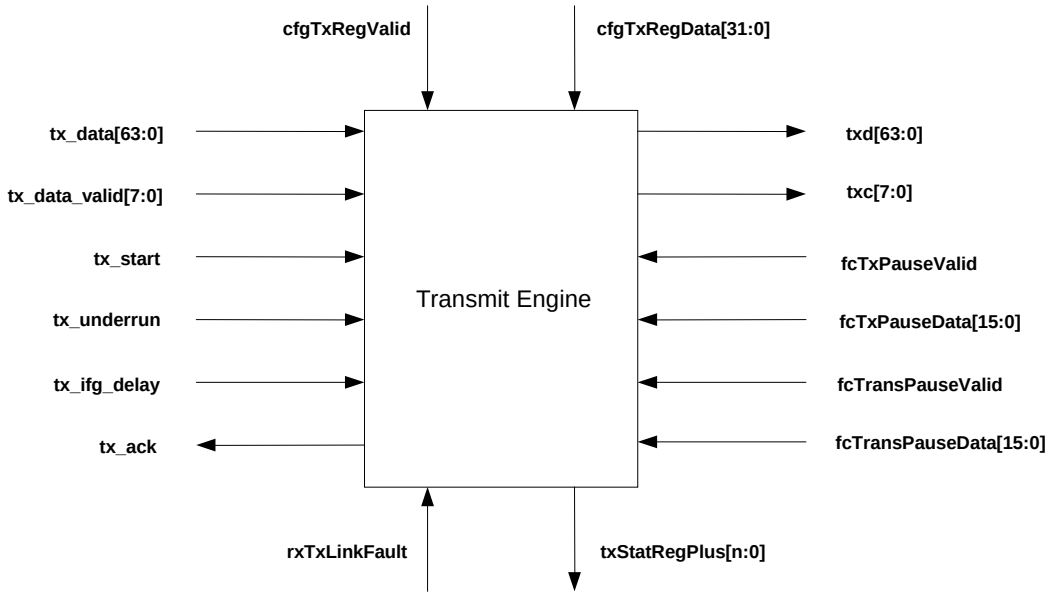

Figure 2-1 Diagram of the Transmit Block

2.2 Module Interface

Table 2-1 lists the I/Os that interface to the client.

Port Name	Direction	Description
tx_data[63:0]	Input	Frame data to be transmitted is supplied on this port.
tx_data_valid[7:0]	Input	Control signals for tx_data port. Each asserted signal on tx_data_valid signifies which bytes of tx_data are valid; i.e., if tx_data_valid [0] is '1', the signals tx_data [7:0] are valid.
tx_start	Input	Handshaking signal. Asserted by the client to make data available for transmission
tx_underrun	Input	Assert this pin to forcibly corrupt the current frame
tx_ifg_delay[7:0]	Input	Control signal for configurable inter-frame gap adjustment

tx_ack	Output	Output Handshaking signal. Asserted when the first column of data on TX_DATA has been accepted

Table 2-1 Client-side interface

Table 2-2 lists the I/Os that interface to the physical.

Port Name	Direction	Description
txd[63:0]	Output	Output. Transmit data to PHY. It transfers data frame in such format:
txc[8:0]	Output	Output. Transmit control to PHY. Figure 2.4 and Figure 2.5 can illustrate its function

Table 2-2 Physical-side interface

Table 2-3 lists the I/Os that interface to the flow control.

Port Name	Direction	Description
fcTransPauseData[15:0]	Input	MAC Control Frame data to be transferred
fcTransPauseVal	Input	when it is assert, it indicates fcTransPauseData is valid.
fcTxPauseData [15:0]:	Input	Cycles for Transmit Engine state machine to pause
fcTxPauseValid	Input	when it is assert, it indicates that transmit state machine should pause for cycles indicated by fcTxPauseData

Table 2-3 Flow control interface

Table 2-4 lists the I/Os that interface to the MDIO.

Port Name	Direction	Description
cfgTxRegData[31:0]	Input	The value from configuration registers
CfgTxRegValid	Input	When it is asserted, it indicates cfgTxRegData is valid
txStatRegPlus[n-1:0]	Output	n presents the number of statistic registers. Each bit presents add operation to a statistic register. These signals should only last for one cycle. For example, when txStatRegPlus [0] is asserted for one cycle, the counter of Control Frames Transmitted OK register in Management Module will plus one

Table 2-4 MDIO interface

Table 2-5 lists the I/Os that interface to the Receiver.

Port Name	Direction	Description
rxTxLinkFault	Input	Indicate that Receive Engine has received Local Fault. Generate IDLEs at the PHY.

Table 2-5 Receiver interface

2.3 Module Design

The transmit engine contains several blocks; input and output FIFO/register, control logic and counters. The input and output FIFO/registers are employ to receive data from the client and distribute the data to the physical. All data flow is all under controlled from the control logic.

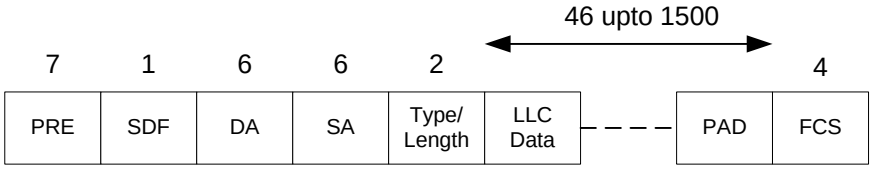

Figure 2-2 IEEE 802.3x Frame

Figure 2-7 shows the standard MAC frame used for transmitting to the physical. Preamble has a value of 10101010 and the SFD has a value of 10101011.

Appendix A shows the different data supplied by the client-side.

The tx_ack signal is generated using a type of counter circuitry to compensate when paused frame transmission is invoked by the flow control block or the inter frame delay is set at the start. The assertion of the signal is achieve when the count equal to the delay value. The request from the pause or inter frame will used to select the counter delay value.

The minimum inter frame gap is 96 bits. For a normal transmission, the delay value will be 2 clock cycles as shown in Appendix A.

Table 2-1 shows the Transmitter Configuration Word that is send by the Management block.

Bit	Default Value	Description
24-0	N/A	Reserved
25	0	Inter-Frame Gap Adjust Enable. When this is set to 1, the transmitter will read from the port tx_ifg_delay at the start of the frame transmission and adjust the interframe gap accordingly. When this set to bit 0, the transmitter will output the minimum interframe gap.
26	0	WAN mode enable. When this is set to 1, the transmitter will automatically insert extra idles into the interframe gap to reduce the average data rate that of the OC-192 SONET payload rate (WAN mode). When this bit is set to 0, the transmitter will use the normal Ethernet interframe gap. **Not supported.**
27	0	VLAN Enable. When this bit is set to 1, the transmitter will allow the transmission of the VLAN tagged frames.
28	1	Transmitter enable. When this bit is 1, the transmitter is operational. When this is 0, transmitter is disabled.
29	0	FCS enabled. When 1, the data will be supplied with FCS. When bit 0, the MAC will append padding when required and computes the FCS field.
30	0	Jumbo Frame enable. When this bit is set to 1, the frame size greater than 1518 bytes is accepted. (9000 bytes) When bit 0, the transmitter can only accept bytes upto 1518 bytes.
31	0	Transmitter reset. When this is set to 1, the transmitter is reset and automatically revert to 0. Reset will set configuration to default values.

Table 2-6 Transmitter Configuration Word

The control logic is essentially a state machine that controls how the data is output to the physical by selecting between the control bytes and the client data. There are four different states in the control logic and there are IDLE, START, DATA and PAUSE.

In the IDLE state, IDLE bytes (07) are transmitted to the physical. When a receive fault occurs, the state machine will be stuck at IDLE until the signal is de-asserted.

In the START state, START control bytes, PREAMBLE bytes and Start Frame Delimiter are loaded into the output. Once the data is loaded, the state changed to DATA.

In the DATA state, the FIFO empty 64 bits at a time to the output until the empty flag is set. In this state, the tx_data_valid bytes are inverted and output to the command output, txc.
If the empty flag is asserted in any of the FIFO, the output register with no valid data will be loaded with the TERMINATE and IDLE control bytes. After the last frame is transmitted, the state machine will either change to the IDLE state or when fcTransPauseVal signal is asserted, the state machine will change to the PAUSE state.

In the PAUSE state, a pause frame is sent out to the physical. In this state, the tx_ack is delayed until the pause frame is transmitted. The tx_ack signal is used to indicate to the client that the first data column is received.

Figure 2-3 shows the format for a pause frame. The OPCODE value is 0001.

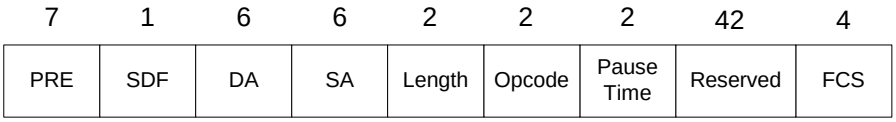

Figure 2-3 PAUSE Frame

When the tx_underrun is asserted by the client-side, an error code will be inserted to the frame transmission.

When VLAN is enabled, the transmitter is able to accept VLAN frames. Figure 2-4 shows the VLAN frames. The VLAN ID has a value of 8100 and the total frame size if extended to 1522 bytes.

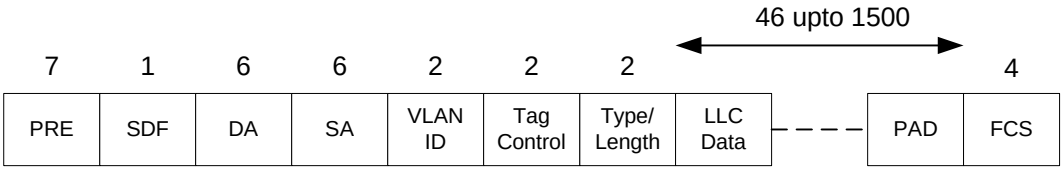

Figure 2-4 Virtual LAN Frame

When FCS is required for the data from the client, a parallel scheme is employed to generate the FCS. If the data is less than 46 bytes, padding is applied to the appropriate FIFO or registers. This is only achieved when the length of frame is received. The length will indicate if the frame is below the minimum frame size. By knowing the size, the appropriate FIFO or register can be applied with the padding.

If the frame is greater than the maximum size, a counter is used to track the number of data columns and using the length field, this will determine when to truncate the data. An error code will be inserted to the transmission to corrupt the frame.

Table 2-1 shows the control characters for the physical. In the txc output, bit 1 indicates a non-data being transmitted at txd and bit 0 for data transmission.

TXC	TXD	Description
0	00 through FF	Normal Data Transmission
1	07	Idle in ||I||
1	07	Idle in ||T||
1	9C	Sequence
1	FB	Start
1	FD	Terminate
1	FE	Error
1	Any other values	Invalid character

Table 2-7 Control Characters

The command characters are set by inversing the tx_data_valid signals.

For statistic counters, there will be several signals used to send to the txStatsRegPlus[n:0]. The signal has a length of clock cycle. Each signals sent would increment a count within the Management block that keeps a track of the different frames sent.

2.4 Block Diagram

This is a basic block diagram for a normal transmission with FCS supplied.

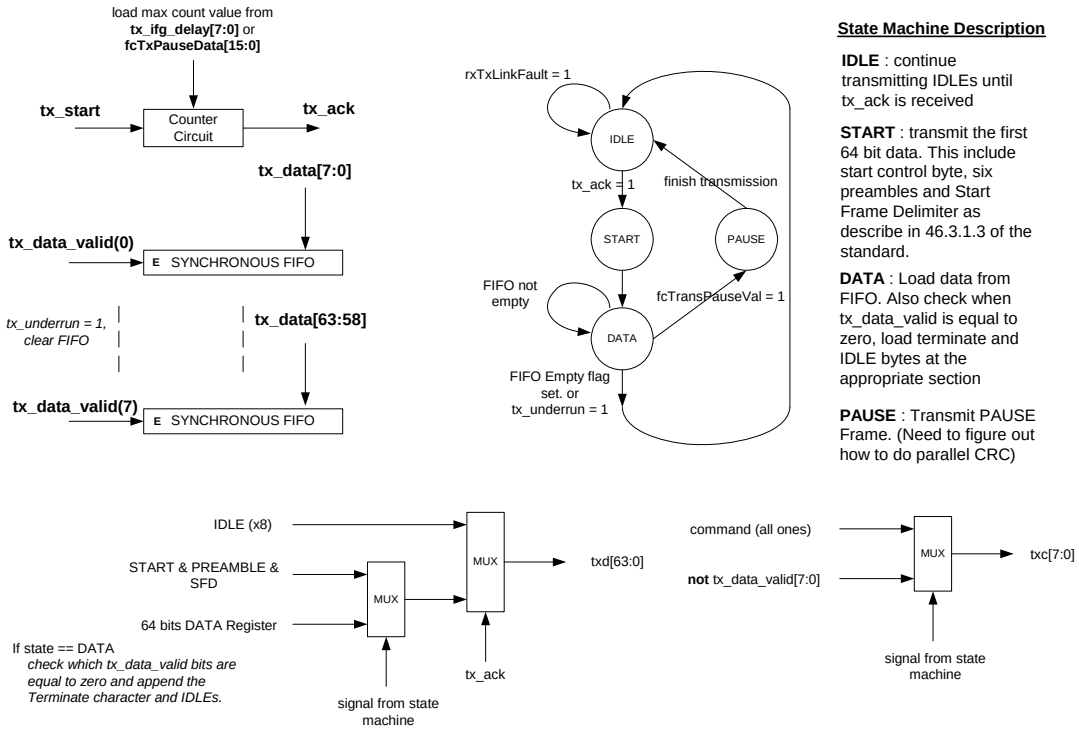

Figure 2-5 Block diagram of the transmit process

Still need to draw more diagrams for the full complete system.

2.5 Code Listing

Source Code Name	Version	Date

Figure 2-6 Code Listing

3 Traceability Matrix

802.3ae Clause	Implemented In

Figure 3-1 Traceability Matrix

4 Abbreviation

FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
PHY	Physical
UML	Unified Modelling Language

Appendix A

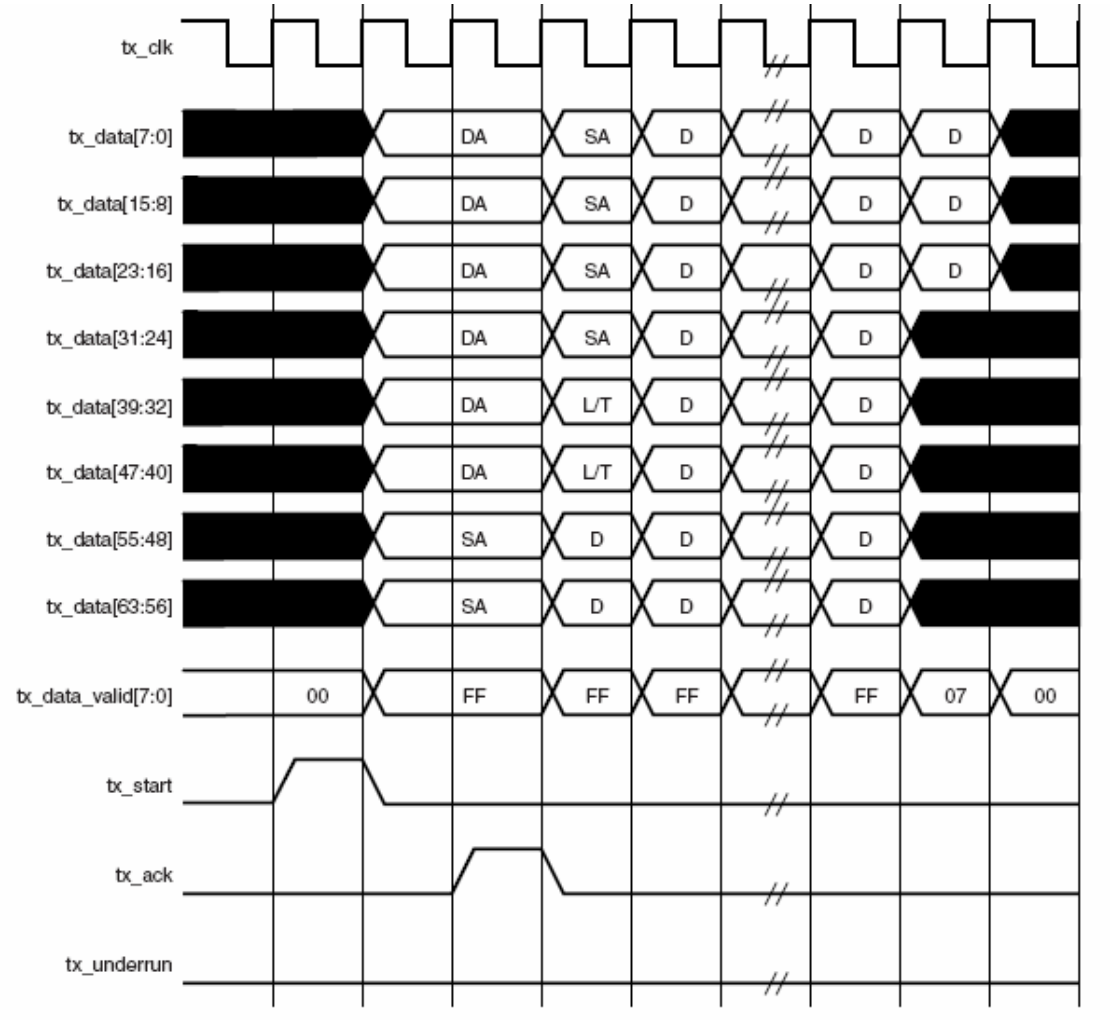

Figure 0-1 Frame Transmission Across Client-Side interface [Ref. 2]

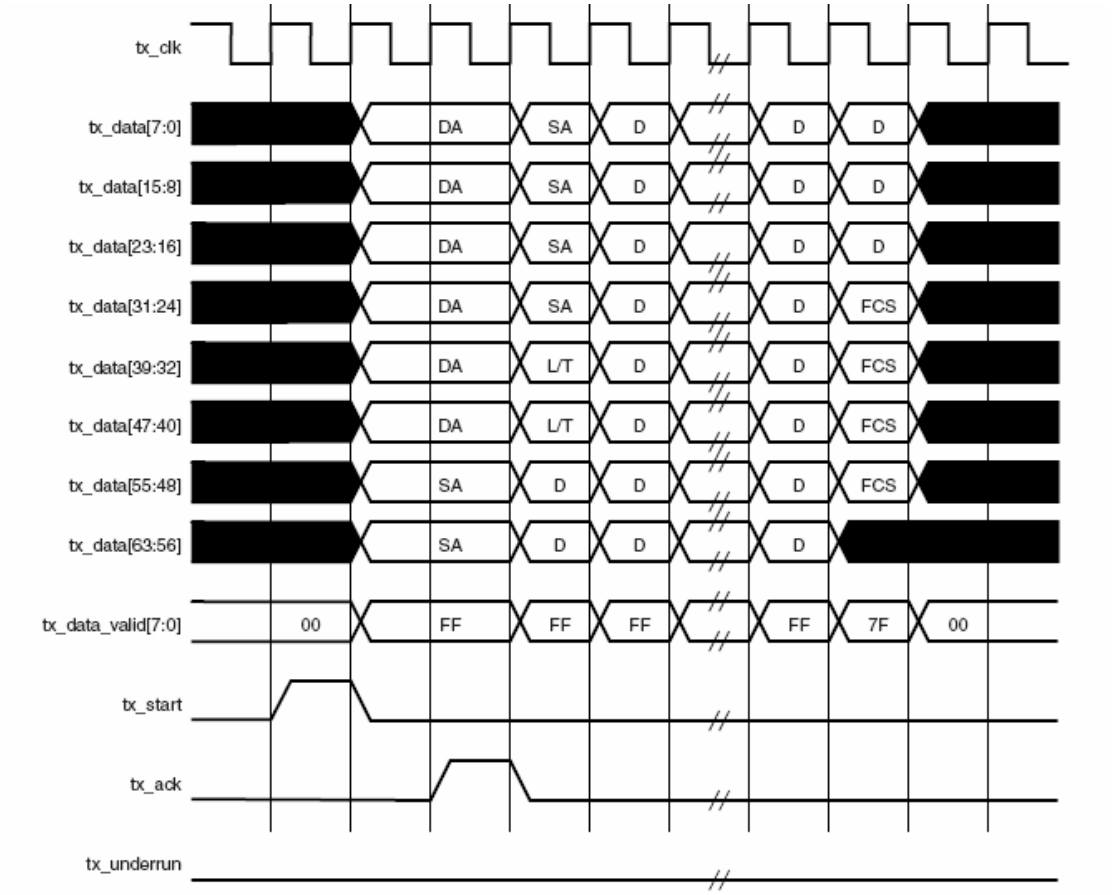

Figure 0-2 Frame Transmission with Client-supplied FCS [Ref. 2]

Figure 2-3 and 2-4 shows a normal transmission from the client-side. FCS has to be generated if it is not included with the data from the client-side. If the data width is below 46 bytes, padding is needed to bring it inline with the minimum frame size. A parallel scheme has to be employ to generate the FCS. (How would I do this and how long would it take)

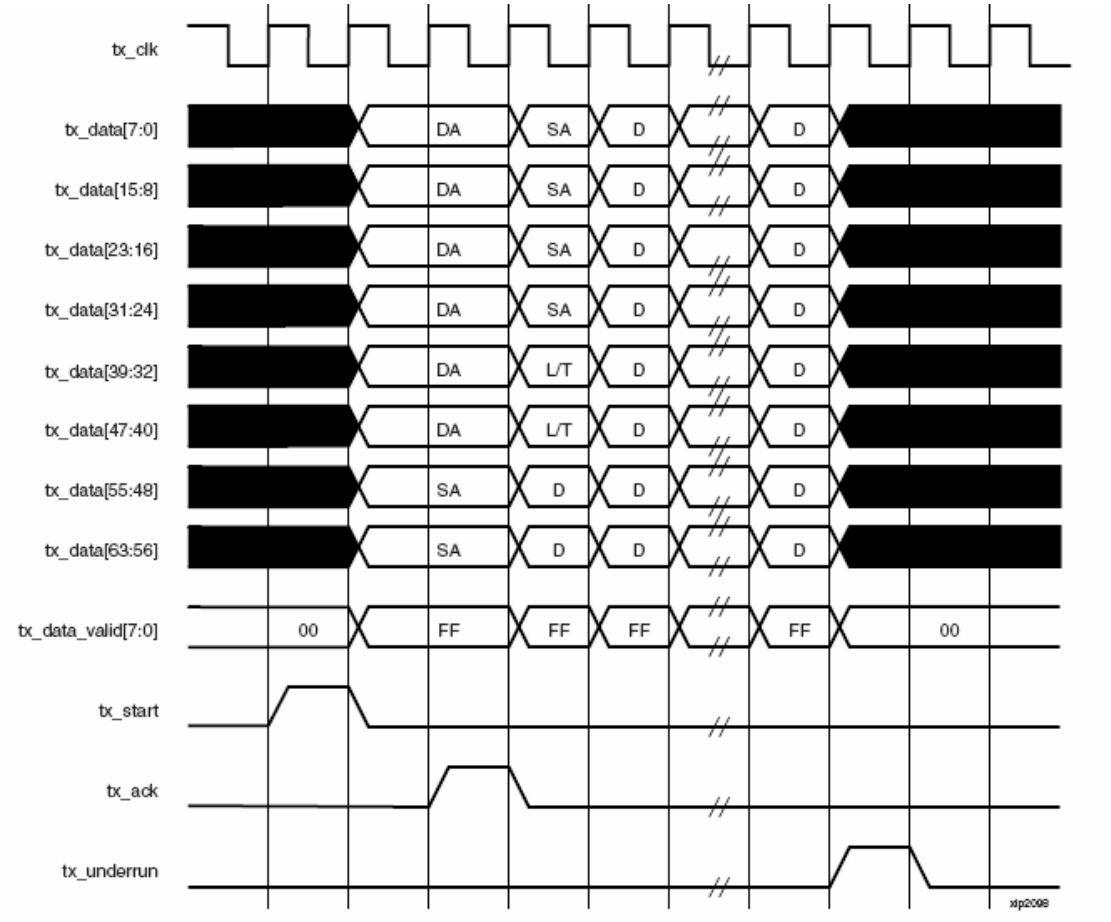

Figure 0-3 Aborting a frame transmission [Ref. 2]

Figure 2-5 shows the abortion of a frame by asserting the tx_underrun signal. The current frame is stopped from transmission. (What happen when is aborted, do I insert an error control character)

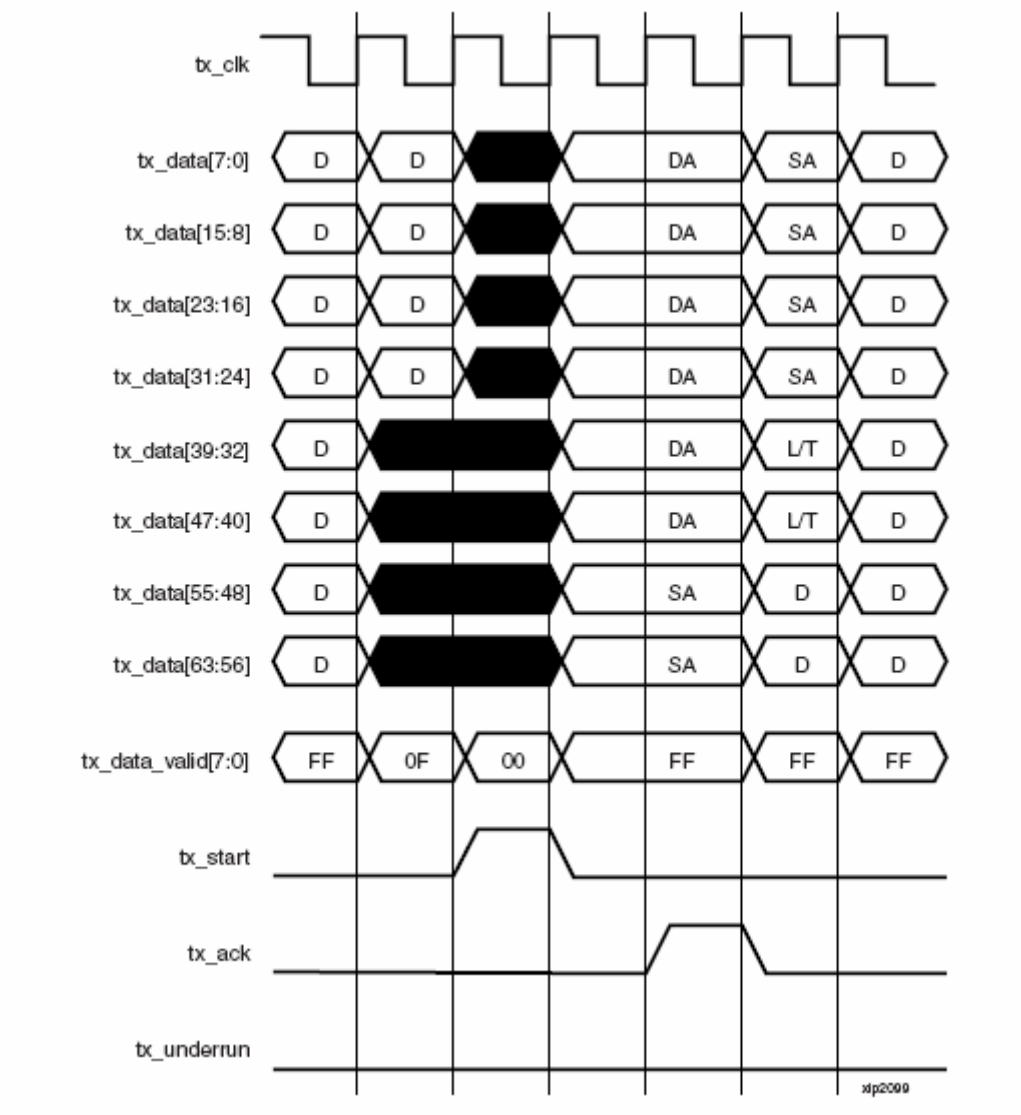

Figure 0-4 Back-to-Back Frame Transmission, no back pressure [Ref. 2]

Figure 2-6 shows a back-to-back frame transmission. An efficient scheme has to be employed to allow this operation.

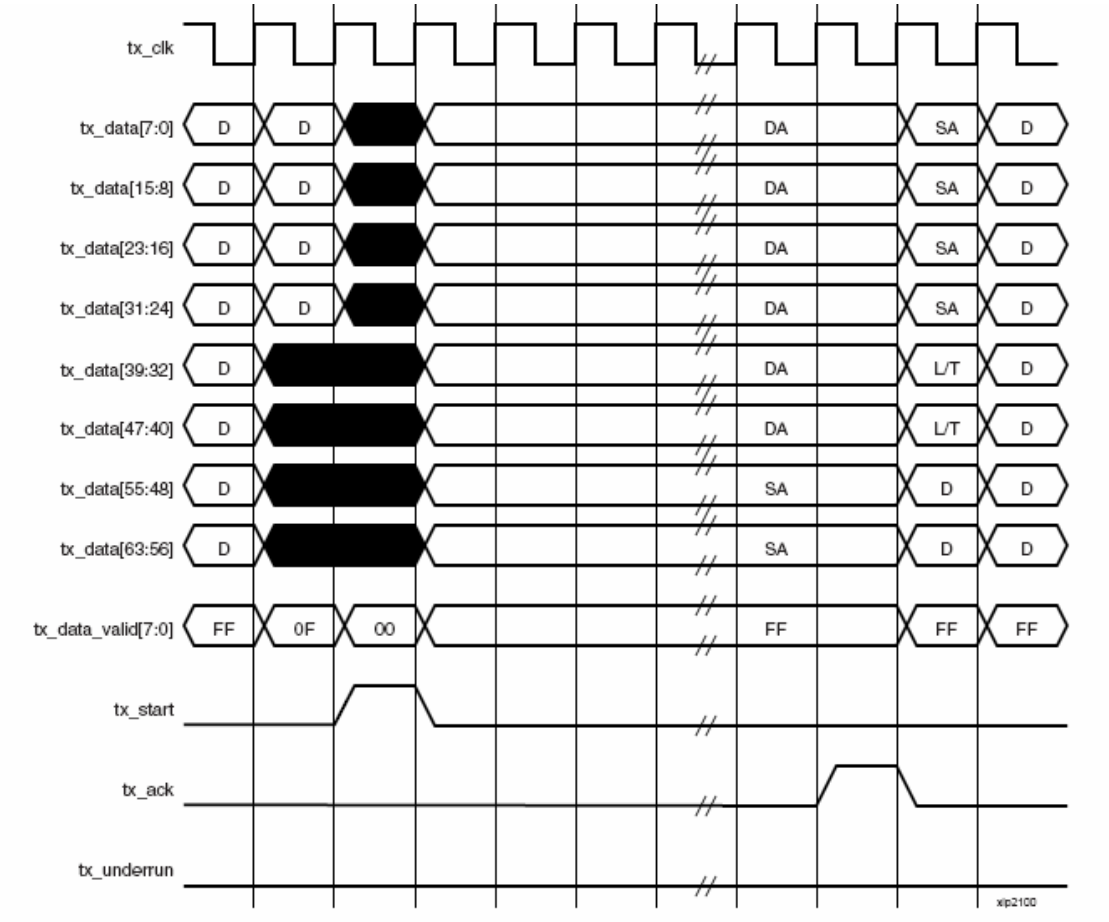

Figure 0-5 Back-to-Back Frame Transmission with Back Pressure from MAC [Ref. 2]

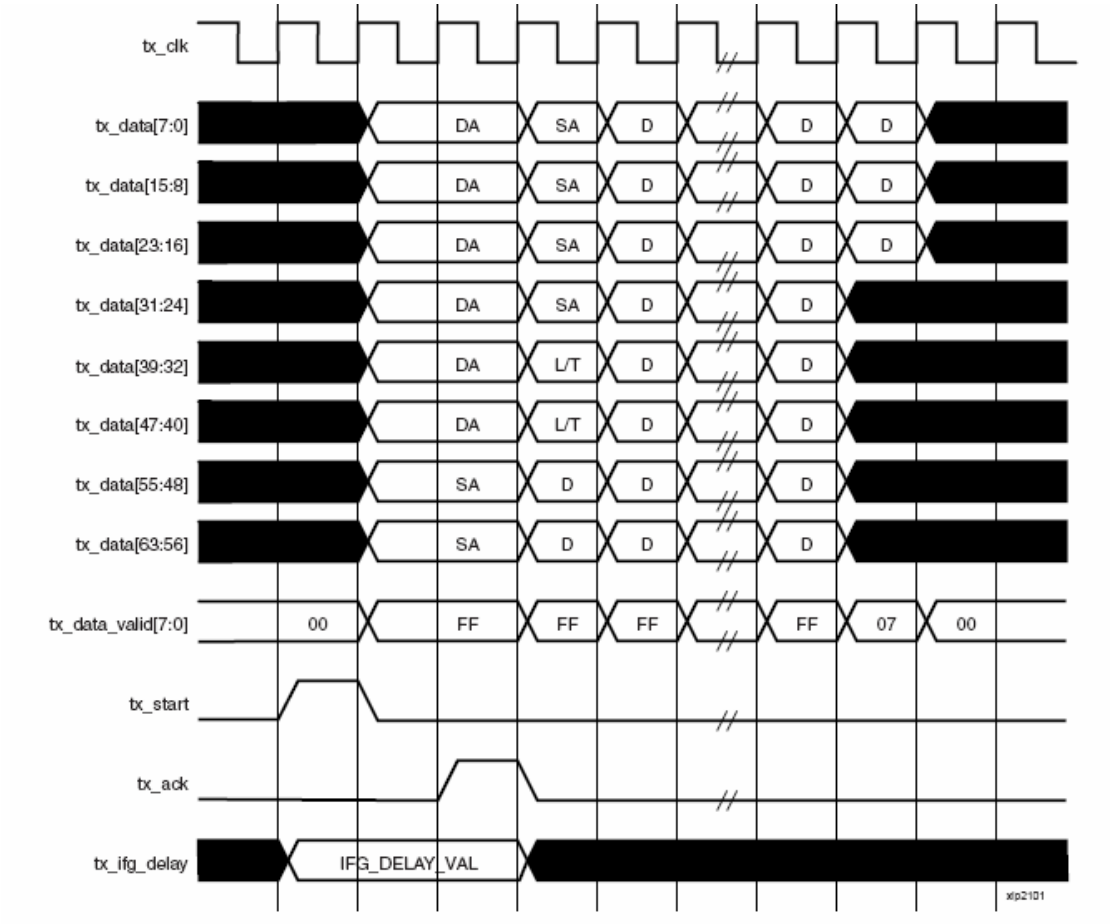

Figure 0-6 Inter Frame Gap Adjustment [Ref. 2]

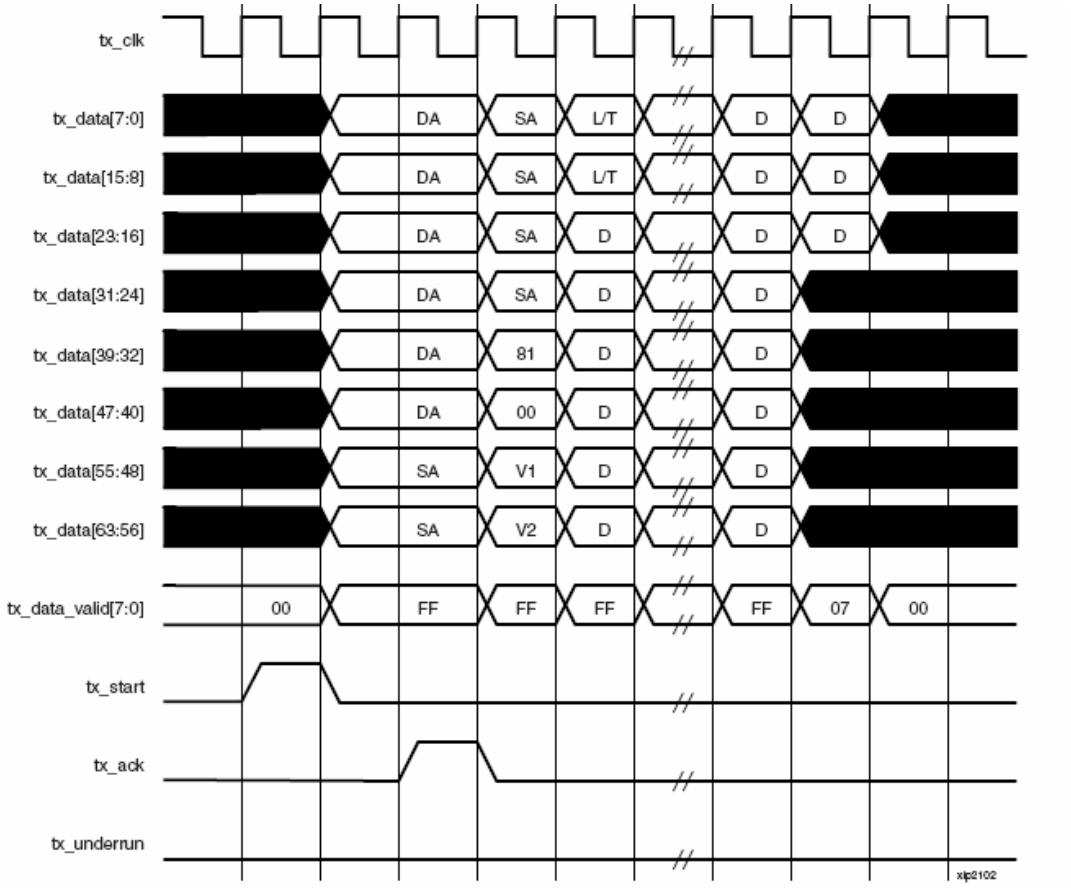

Figure 0-7 Transmission of a VLAN Tagged Frame [Ref. 2]

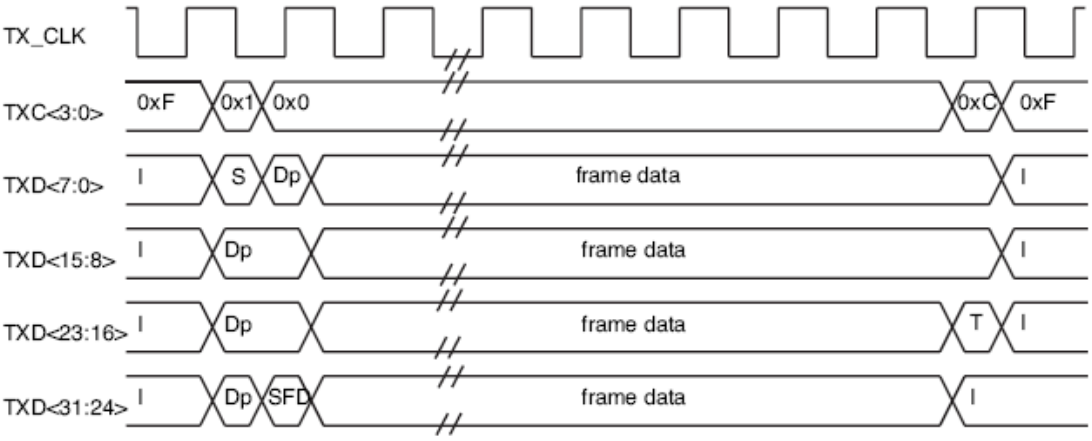

Figure 0-8 Normal frame transmissions [Ref. 3]

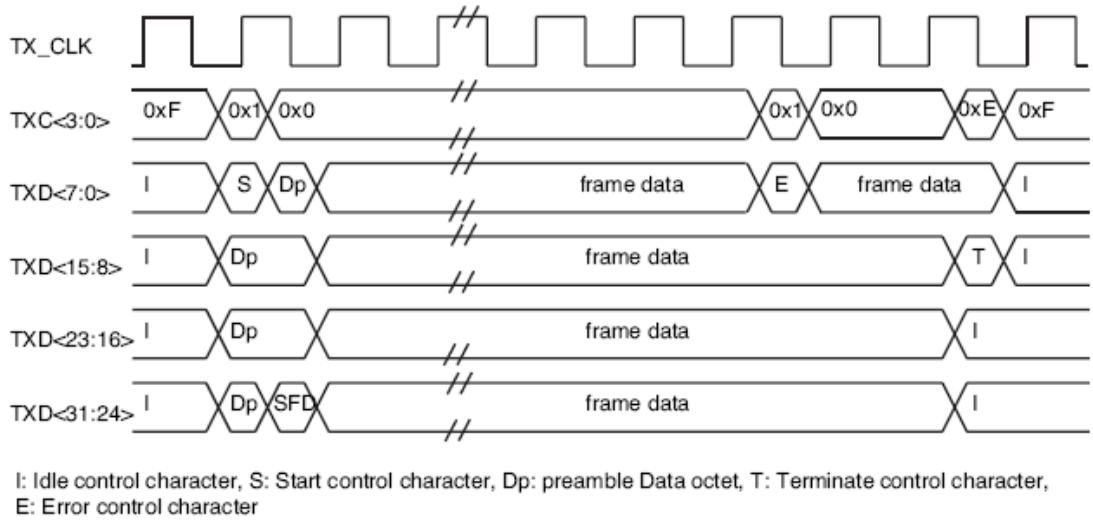

Figure 0-9 Transmit Error Propagation [Ref. 3]