e-Port (HDLC) IP Core

Specifications document

V0.3 DRAFT

Authors

Sandro Bonacini, CERN PH-ESE/ME (sandro.bonacini@cern.ch)
Kostas Kloukinas, CERN PH-ESE/ME (kostas.kloukinas@cern.ch)

Document History

V0.1: Base version.

V0.2: Include PHY description.

V0.3: Include new features (auxiliary port, packet acknowledgement, etc)

1 Introduction

The e-port IP core is an intellectual property (IP) block designed to interface the User Application Logic residing in the Front-End ASICs with the Gigabit Transceiver (GBT) chip via the GBT "e-link" communication lines. It consists of the PHY (PHYsical) hard IP block that implements the "e-link" electrical interface and MAC (Media Access Controller) soft IP block that implements the "e-link" transmission protocol.

2 Features

- Implements the "e-link" transmission protocol, based on the HDLC standard.
- Operation at 80 Mbps scalable up to 320 Mbps using a multilane mechanism.
- Non-deterministic link latency.
- Bi-directional operation.
- Simple Data FIFO interface to user application logic compatible with the Altera ATLANTIC SOC interface.
- Simple control interface providing access to configuration and status registers compatible with the Wishbone SOC interface.
- Unmanaged operation, guaranties communication without requiring the involvement of the user application logic. Acknowledgement/reject of packets provided.
- Auxiliary port for redundant failsafe link.
- SEU-robust design.
- Low-power design.
- Available for ASIC implementation in CMOS 130nm technology.

3 General Description

3.1 Block diagram

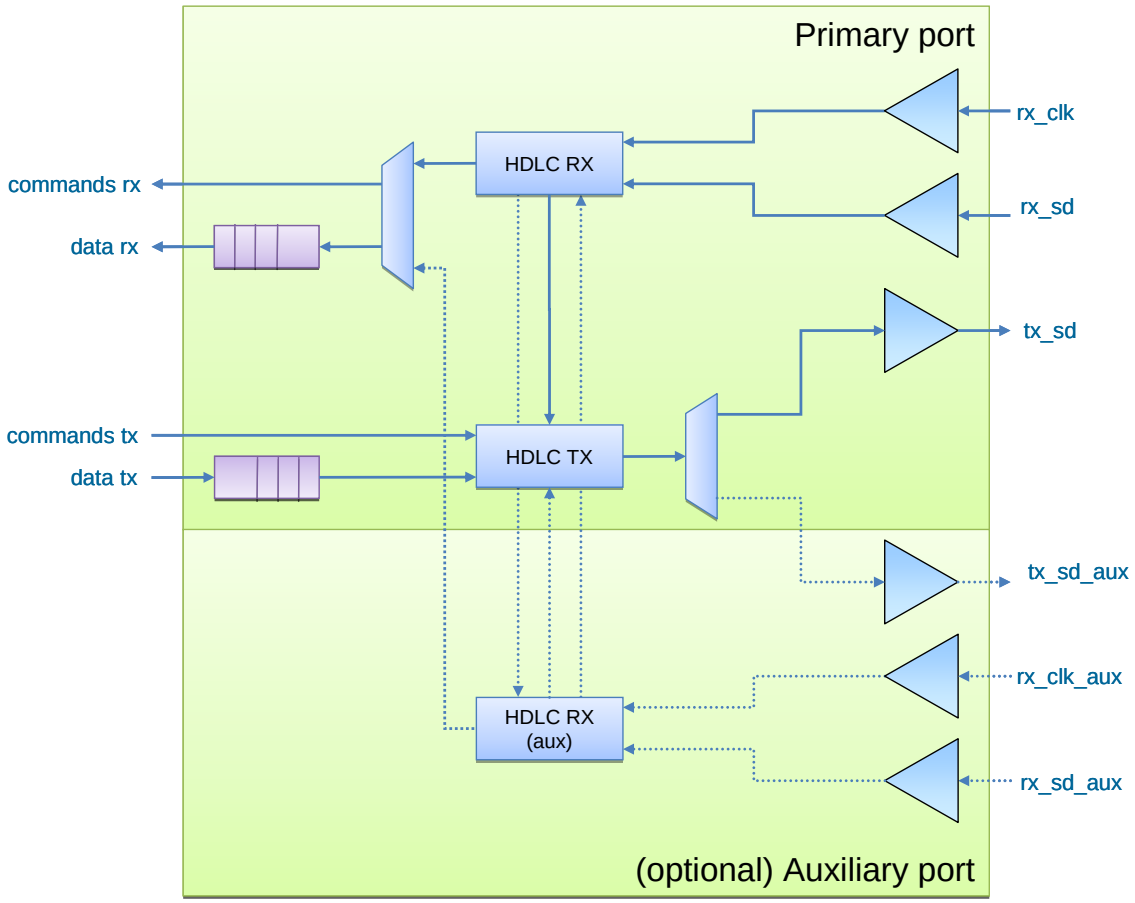


Figure 3-1 The Block Diagram of the e-port HDLC IP Core.

3.2 Parameters

Parameter	Width	Default	Description
HEADER_FIELD	1	1	If true, header field is present in the packet structure. The header field carries address and command information.
MASTER	1	0	If true, port behaves as master, otherwise as slave. A master can send reset and test commands, while a slave receives reset commands and feeds-back test commands. Commands are carried only if header field is present (HEADER_FIELD=1).

ADDR_WIDTH	-	5	Transmit and receive FIFO addressing width.
MAX_PACKET_LENGTH	-	16	Maximum expected packet length
THRESHOLD	ADDR_WIDTH $2^{ADDR\_WIDTH-1}$ – MAX_PACKET_LENGTH		Threshold for asserting the transmit ready to receive data signal *tx_dav*. Must be consistent with ADDR_WIDTH and the size of the packet.

3.3 I/O Signal Description

3.3.1 Global signals

Port	Width	Direction	Description
user_clk	1	Output	User clock
resetb	1	Input	Asynchronous reset

3.3.2 Rx FIFO connections

Port	Width	Direction	Description
rx_ena	1	Output	Transfer enable (data valid)
rx_dav	1	Input	Ready to receive data
rx_sop	1	Output	Start of packet
rx_eop	1	Output	End of packet
rx_cmd_test	1	Output	Test command receive flag
rx_cmd_reset	1	Output	Reset command receive flag
rx_cmd_ua	1	Output	Unnumbered acknowledge receive flag
rx_cmd_srej	7	Output	Reject frame receive flag
rx_ns			
rx_adr	8	Output	Address bus
rx_dat	16	Output	Data bus

3.3.3 Tx FIFO connections

Port	Widt	Direction	Description
tx_ena	1	Input	Transfer enable (data valid)
tx_dav	1	Output	Ready to receive data
tx_cmd_test	1	Input	Send test command flag
tx_cmd_reset	1	Input	Send reset command flag

tx_cmd_sabm	1	Input	Send connect command flag
tx_ns			
tx_adr	8	Input	Address bus
tx_dat	16	Input	Data bus

3.3.4 SLVS I/O connections

Port	Width	Direction	Description
tx_sd	2	Output	Differential serial data trasmission line
rx_sd	2	Input	Differential serial data receive line
rx_clk	2	Input	Differential input clock line
tx_sd_aux	2	Output	Auxiliary differential serial data transmission
rx_sd_aux	2	Input	Auxiliary differential serial data receive line
rx_clk_aux	2	Input	Auxiliary input clock line
tx_cset	4	Input	Transmitter current level setting

4 The e-Port HDLC PHY

The PHY is an IP block implementing serialization/deserialization and symbol synchronization functions. This is done employing the HDLC standard, which uses bit-stuffing techniques in order to achieve symbol synchronization.

An 8-bit frame delimiter flag (binary 01111110) is provided in the standard and the protocol assures that this combination of bits is not found anywhere else in the stream. The frame delimiter contains 6 consecutive ones; therefore any sequence of 5 ones in the stream is stuffed with one following zero at the transmitter side. At the receiver side, every sequence of 5 ones followed by a zero is stripped of the trailing zero.

Any sequence of more than 6 ones is considered to be frame abort or channel idle signaling and resets the receiver PHY state machine. When idle, the transmitter PHY sends repeatedly a fill-frame sequence composed of 7 ones followed by a zero.

The PHY provides serialization/deserialization at double data rate (DDR), therefore the clock frequency is half of the bit rate.

5 The e-Port HDLC MAC

The MAC is a soft IP block implementing the address insertion and retrieval functions, error checking, control command frame generation and execution, data acknowledgment/reject. It connects to the e-port PHY via a private parallel communication bus.

5.1 The HDLC Frame Format

The e-link frame format follows the HDLC standard with some exceptions (ISO/IEC 13239:2002, http://cdsweb.cern.ch/record/442122). The "e-link" data is encapsulated in frames as shown in Figure 5-1. The fields in the frame are transmitted from left to right. The bits within the frame are transmitted from left to right (from least significant bit to most significant bit).

Flag	Address	Control	Information	FCS	Flag
0111110	8 bits	8 bits	16*n bits	16 bits	01111110

Figure 5-1. The "e-link" Frame Format.

The "e-link" frame consists of:

Flag: frame delimiter. The frame delimiter field marks the start and end of the frame and contains a pattern with 6 consecutive ones to which the receiver PHY can synchronize.

Address: destination address. The destination address field is 1 byte in length. During data transmission this field is inserted by the MAC logic based on the content of the *tx_adr* bus. During data reception this field is extracted from the frame by the MAC logic and it is passed to the User Application logic on the *rx_adr* bus.

Control: command field. The control field is 1 byte in length. The value of this field determines if the frame contains information or not, and in the latter case it carries the command code (reset, test, connect, acknowledge, reject). In information frames, the control field carries the transmitted packet number and the last correctly received packet number.

Information: data. The data field may vary in length for a data frame; nevertheless it must be a multiple of 16 bits. This field is not present in case of a command frame.

FCS: Frame check sequence. The FCS field is used to detect transmission errors and is 2 bytes in length. The value of the FCS field is calculated over the address, control and information fields using the CCITT standard 16-bit Cyclic Redundancy Check (CRC) defined as:

$G(x) = x^{16} + x^{12} + x^5 + 1$

5.2 Functional description

The HDLC e-port serializes the *tx_dat* parallel input bus onto the tx_sd pin output and deserializes the rx_sd pin input onto the *rx_dat* parallel output bus.

The interface to the *tx_dat* and *rx_dat* buses is synchronous with the system clock user_clk, obtained by the receiver interface. Two FIFO memories are provided in the e-port block in order to store the received data and the data to be transmitted. The FIFO interfaces follow the Atlantic interface functional specification.

The system clock *user_clk* is rising-edge active, using single-edge clocking.

5.2.1 Parallel data interface (Atlantic interface)

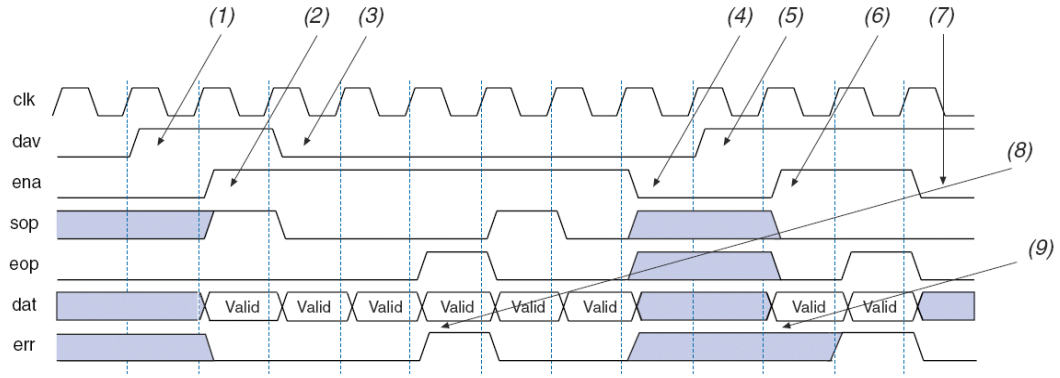
The data interface to the host logic adopts the ALTERA Atlantic interface with some minor exceptions which will be listed in the following sections.

5.2.1.1 Receive interface

The e-port acts as master for the receive interface, following the Atlantic interface functional specification.

The e-port (master) asserts *rx_ena* and drives new data on *rx_dat* and a corresponding packet address on *rx_adr*. The slave observes *rx_dat* and *rx_ena*. If *rx_ena* is asserted, the slave accepts and processes data; if not, it discards *rx_dat*. If *rx_ena* is deasserted, *rx_dat* and *rx*_adr are undefined. The slave sink has no cycle-by-cycle flow control; it uses *rx_dav* to request the e-port (master) to stop data transfer. However, the e-port (master) may take *threshold* clock cycles to stop transferring data, depending on the application. *rx_dav* indicates that the slave can accept *threshold* data words.

The following figure depicts a timing diagram of the interface protocol.



Notes:

(1) Slave sink indicates it has space for threshold words.

(2) The master source begins writing data to the slave sink.

(3) Slave sink indicates it no longer has space for threshold words. Master source can continue to send data, but

must ensure that the slave sink does not overflow.

(4) Master source stops sending data.

(5) Slave sink indicates it has space for threshold words.

(6) Master source begins writing data to the slave sink.

(7) Slave sink indicates it still has space, but the master source has run out of data.

(8) Master source detects an error, and asserts err.

(9) Master source identifies a data error, it asserts err and holds it until eop is deasserted.

The *rx_sop* and *rx_eop* signals are used to delineate the packet boundaries on the *rx_dat* and *rx_adr* buses. The *rx_adr* bus remains valid and constant during an entire packet.

The *rx_err* signal is used to indicate that the current packet is aborted and should be discarded. *rx_err* can be asserted at any time during the current packet, but once asserted it can only be deasserted on the clock cycle after *rx_eop* is asserted. *rx_err* is normally used in case of checksum mismatch, buffer overflow, parity error or abort sequence detection.

For more information about the Atlantic interface specifications, please consult http://www.altera.com/literature/fs/fs_atlantic.pdf .

NOTE: (non-standard) Upon initiation of a packet delivery the master receive interface will continue to the end of the packet without pause. A pause (*rx_ena* deasserted) in the middle of a packet is not so far supported.

5.2.1.2 Transmit interface

The e-port acts as slave for the transmit interface, conversely to the receive part, following the Atlantic interface functional specification. Nevertheless, the start of packet (*sop*), end of packet (*eop*) and error (*err*) signals are not present since not necessary for the transmit logic. The interface needs in fact at least an idle cycle between consecutive packets in order to insert address and control information. The *tx_dav* signal is used in the handshake to force this condition.

The master asserts *tx_ena* and drives new data on *tx_dat* and a packet address on *tx_adr*. The e-port (slave) observes *tx_dat* and *tx_ena*. If *tx_ena* is asserted, the e-port (slave) accepts and processes data; if not, it discards

tx_dat. If *tx_ena* is deasserted, *tx_dat* and *tx_adr* are undefined. The e-port (slave sink) has no cycle-by-cycle flow control; it uses *tx_dav* to request the master to stop data transfer. However, the master may take *threshold* clock cycles to stop transferring data, depending on the application. *tx_dav* indicates that the e-port (slave) can accept *threshold* data words. If the master continues to assert *tx_ena* for an extended period of time after *tx_dav* is deasserted, the e-port internal transmit FIFO may overflow.

The timing diagram for the transmit interface is identical to the one for the receive interface.

NOTE: (non-standard) Upon initiation of a packet delivery the slave transmit interface expects the master to continue to the end of the packet without pause. A pause (*tx_ena* deasserted) in the middle of a packet is not supported.

5.2.2 Serial data interfaces (primary and auxiliary)

The HDLC block features two serial data interfaces, a primary interface and an (optional) auxiliary interface. The two interfaces can be connected to two separate GBTX chips for redundancy. If one of the two optical links is broken, it is still possible to access the front-end via the other interface.

Upon startup it is necessary to initialize the desired interface for use with a reset command. Only one interface can be used at a time, data from a non-initialized interface will be discarded. It is possible to switch from one interface to the other with the connect (SABM) command.

The serial interfaces use the SLVS signaling standard.

5.2.3 Commands and responses

The MAC provides commands for the initialization and control of the slave HDLC block from the master. Only a few commands of the HDLC standard are implemented. These are (in order of priority): RSET (reset), SABM (connect), TEST, SREJ (selective reject).

Upon reception of the reset and connect commands, the slave generates an UA (command acknowledge) response. The TEST command generates a TEST response.

The behavior of the MAC is controlled by the MASTER parameter. Only the master can send reset, test and connect commands, while only the slave can send SREJ commands. The MAC can send only one command at a time, therefore in case several commands are asserted by the user logic in a short time, only one command will be sent. It is safer to wait for the response before sending the next command.

5.2.3.1 Synchronous reset command (RSET)

A master transmitter can send synchronous reset commands to the slave receiver. For this purpose the *tx_cmd_reset* signal on the master transmitter side should be asserted for one clock cycle. The transmitter generates a reset command frame which has priority over all other frames. Upon reception of the reset command frame, the slave receiver asserts the *rx_cmd_reset* signal which resets the MAC circuitry and can be used as asynchronous reset for the user logic. The reset command generates an UA (unnumbered acknowledge) response.

Upon reception of the reset command, the port (primary or auxiliary) that receives the reset is considered to be active and the other interface is disconnected.

5.2.3.2 Connect command (SABM)

The SABM (Set Asynchronous Balanced Mode, here referred as "connect") command instructs the MAC on which port (primary or auxiliary) is active. The port (primary or auxiliary) that receives the command is considered to be active and the other interface is disconnected.

For this purpose the *tx_cmd_sabm* signal on the master transmitter side should be asserted for one clock cycle. The connect command generates an UA response.

5.2.3.3 Test coomand (TEST)

The loopback test function can be useful for link verification purposes. The loopback test is controlled on the master transmitter side by the *tx_cmd_test* signal which, when asserted for one clock cycle, instructs the transmitter MAC to generate a test command frame. At the slave side, the test command is received and looped-back to the slave transmitter, which sends it back to the master. Upon reception of the test command frame, the master asserts the *rx_cmd_test* signal. The RX and TX FIFOs are not part of the loopback path.

5.2.4 Packet numbering and acknowledgement

The HDLC TX block inserts an incremental transmitted packet number (NS) as well as the last correctly received packet number (NR) in the control field of each transmitted frame. The packets are numbered from 0 to 7, therefore there can be a maximum of 8 uniquely numbered packets in the link.

The master shall wait for the acknowledgement of sent packets before sending new ones in order to avoid having two packets in the link with the same frame number.

The HDLC RX block checks every received packet number (NS) against its internal last correctly received packet number (VR) and flags a SREJ command in case packets are missing. CRC-failing packets are dropped and therefore treated as missing packets.

The SREJ command (selective-reject) is sent from the slave to the master and contains the numbers (NS) of the lost packets. The SREJ command contains always a range specification, its format is (as seen in the module port *rx_cmd_srej*):

Flag (1)	Start NS	End NS
1 bit	3 bits	3 bits

Upon reception of a SREJ command the master can decide whether to resend the lost information. The master cannot send SREJ commands to the slave.

5.2.5 Packet address

The packet address buses *tx_adr* and *rx_adr* are enabled only when the header field is present (HEADER_FIELD=1), since the packet address is stored in the header. In case the header field is not present, the received address *rx_dat* is fixed at zero and the transmitted address *tx_adr* is ignored.

5.2.6 Unsupported HDLC features

The elink PHY/MAC is based on the HDLC standard (ISO/IEC 13239:2002) and supports only a limited number of its features:

only the basic frame format is supported;

only Asynchronous Balanced Mode (ABM) is supported;

only the TEST, RSET, SABM, UA and SREJ commands are implemented;

the poll/final bit is ignored and set to zero;

Moreover, if the parameter HEADER_FIELD is set to zero/false, the address and control fields are not present, therefore no commands are implemented and the frames are by default unnumbered information frames.