ESIstream

The **E**fficient **S**erial **I**nterface

ESIstream 62B64B - Protocol Specification

Version 1.0

ESIstream is license-free high efficiency serial interface protocol based on 14b/16b encoding.
Its main benefits are low overhead and ease of hardware implementation.

TABLE OF CONTENTS

FIGURES

Documents Amendment

Issue	Date	Comments
1.0	March 2021	Creation

Terminology

ADC	Analog to Digital Converter
ASIC	Application-Specific Integrated Circuit
CDR	Clock and Data Recovery
DAC	Digital to Analog Converter
ESS	ESIstream Synchronization Sequence
FAS	Frame Alignment Sequence
FPGA	Field Programmable Gate Array
LD	Logic Device
LFSR	Linear Feedback Shift Register
PAS	PRBS Alignment sequence
PL	Programmable Logic
PLL	Phase Locked Loop
PRBS	Pseudo-Random Binary Sequence
RX	Receiver
TX	Transmitter
UI	Unit Interval (period of time to send a bit through the serial lane)
Xcvr	Transceiver

1. ESISTREAM 62B/64B PROTOCOL

1.1 Overview

ESIstream provides an efficient High-Speed serial interface based on a 62B/64B encoding using a Linear Feedback Shift Register (LFSR) scrambling unit, a Disparity Bit (DB) to ensure deterministic DC balance transmission and a toggling bit, the Clk Bit (CB), to enable synchronization monitoring.
It is **license-free** and supports in particular serial communication between FPGAs and High-Speed data converters.
However, ESIstream can be used in any system requiring a serial interface. For instance, between two FPGAs or two ASICs.

An ESIstream system is made up of the following elements.

- A transmitter (TX) can be an ADC or any Logic Devices (LD) such as a FPGA or an ASIC.
- A receiver (RX) can be a DAC or any Logic Devices such as a FPGA or an ASIC.
- A number of serial lanes (L ≥ 1) to transmit serial data.
- A synchronization signal (sync) used to initialize the communication and synchronize the transmitter and receiver. On a single device, only one occurrence of the SYNC signal is necessary between the transmitter and the receiver even if multiple serial links are implemented.

There is no clock lane in a serial interface. For each lane, the receiver should recover the clock from the data.

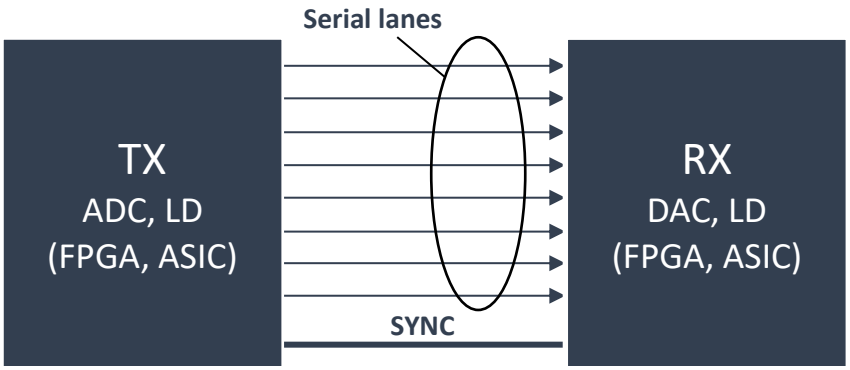

Figure 1: Basic ESIstream system

1.2 TX and RX architecture

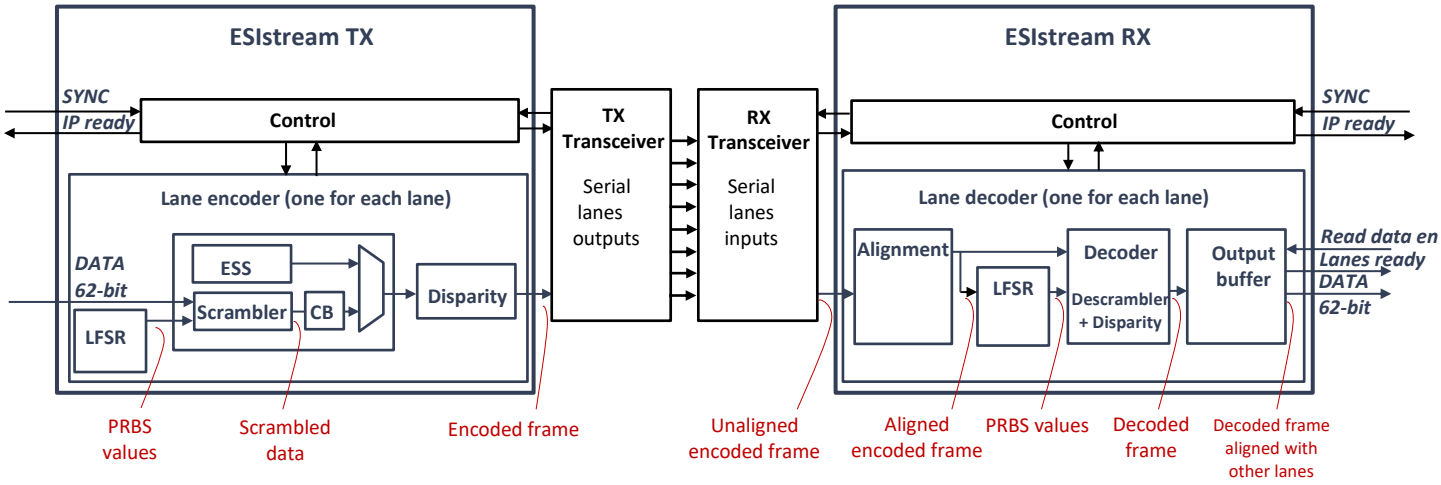

Figure 2: ESIstream TX and RX architecture overview

Different stages of encoding/decoding are implemented to manage the limitations of a serial interface.

An AC coupled interface requires a DC balance, otherwise the AC coupling link will tend to drift and the received data will be corrupted.

The Clock Data Recovery (CDR) in the RX transceiver stage, which usually contains a Phased Locked Loop (PLL), specifies a maximum number of Unit Interval (UI, time to send a bit through the serial lane) without transition on the serial lane, the maximum run length. Otherwise, the PLL can lose its lock.

1.3 Frame

An ESIstream 62B/64B frame is 64-bit wide. The frame is **transmitted**, **LSB first**, between the transmitter and the receiver.
62-bit of useful data are scrambled and then **2-bit of header** are concatenated to create the **64-bit** ESIstream frame.
The **header** is composed of:

- The Clk Bit (CB), which is toggling between each consecutive frame sent through a single serial lane. The Clk bit can be used to monitor the synchronization of a single lane.
  The Clk bit can be disabled and set to 1 to reduce the maximum run length to 64.
- The Disparity Bit (DB), which is the result of a calculation, the disparity processing, done on the 62-bit data and on the Clk Bit.

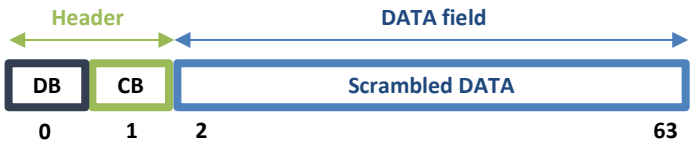

Figure 3: ESIstream 62B64B encoded frame

These different stages of encoding are realized to manage the limitations of a serial interface. First, an AC coupled interface between transmitter and receiver implies that the transmission be DC balance, otherwise the AC coupling capacitor will drift and the received data will be corrupted. Secondly the CDR in the reception stage usually contains a PLL. This means that there must be transitions in the transmission otherwise this PLL will lose its lock.

- When the Clk bit is toggling, the scrambling, the clk bit and the disparity processing ensure a deterministic transmission with a DC balance between **+/- 64** and a max run length of the transmission of **128.**
- When the Clk bit is set to 1, the scrambling, the clk bit and the disparity processing ensure a deterministic transmission with a DC balance between **+/- 64** and a max run length of the transmission of **64.**

Current FPGA CDR can work with a maximum run length of 200 and above.

1.4 Scrambling

Scrambling ensures a statistical DC balanced transmission. It also statistically ensures that there are transitions in the transmission.

ESIstream uses an additive scrambling to avoid error propagation in case of a single bit error. The Linear Feedback Shift Register (LSFR), which generates a Pseudo Random Binary Sequence (PRBS), is based on a Fibonacci architecture and uses the polynomial $X^{31}+X^{28}+1$. It has a length of 2^{31} - 1. See Annex A for more information.

The ESIstream LFSR is characterized by the following equations:

X30 = X27 xor X30
X29 = X26 xor X29
X28 = X25 xor X28
X27 = X24 xor X27
X26 = X23 xor X26
X25 = X22 xor X25
X24 = X21 xor X24
X23 = X20 xor X23
X22 = X19 xor X22
X21 = X18 xor X21
X20 = X17 xor X20
X19 = X16 xor X19
X18 = X15 xor X18
X17 = X14 xor X17
X16 = X13 xor X16
X15 = X12 xor X15
X14 = X11 xor X14
X13 = X10 xor X13
X12 = X9 xor X12
X11 = X8 xor X11
X10 = X7 xor X10
X9 = X6 xor X9
X8 = X5 xor X8
X7 = X4 xor X7
X6 = X3 xor X6
X5 = X2 xor X5
X4 = X1 xor X4
X3 = X0 xor X3
X2 = X27 xor X30 xor X2
X1 = X26 xor X29 xor X1
X0 = X25 xor X28 xor X0

The PRBS is applied to the data with a bitwise XOR binary operation:

DATA[61:0] **XOR** (PRBS2[61:0] & PRBS1[61:0]) = DATA_SCRAMBLED[61:0]

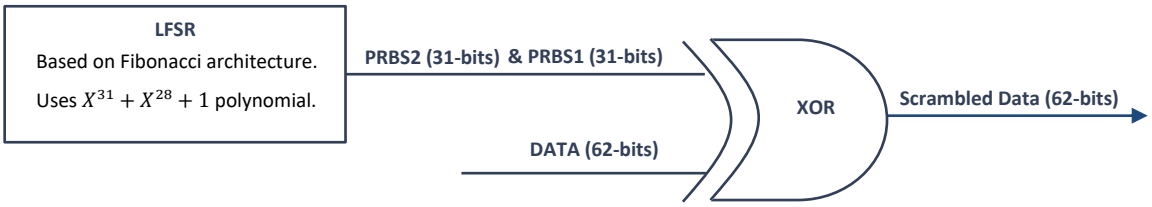

Figure 4: Scrambling principle, bitwise XOR binary operation

In case of a multi-lane interface, in order to reduce correlation between lanes, each lane should have different initial values for the scrambling units.

1.5 Encoding

Scrambled data (62-bit) are encoded are encoded into a 64-bit frame adding two header bits, the Clk Bit (CB) and the Disparity Bit (DB).

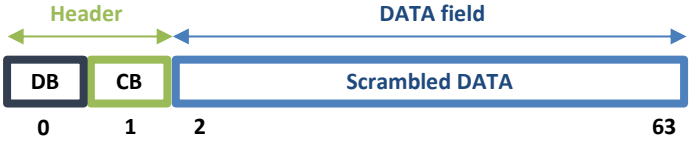

Figure 5: ESIstream 62B64B encoded frame

1.5.1 Clk Bit (CB)

On each serial lane, the Clk Bit (CB) toggles at every ESIstream frame sent through one serial lane.
The receiver uses the Clk Bit to monitor the link synchronization. If the receiver does not detect that the Clk Bit is toggling properly, then it can state that the link is not synchronous or has lost its synchronization and restart the synchronization process.

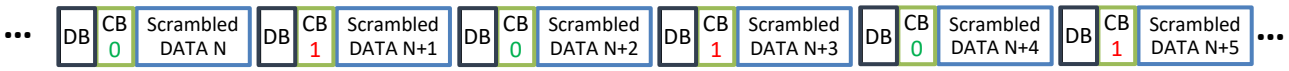

Figure 6: Clk Bit toggling properly on one serial lane.

1.5.2 Disparity Bit (DB)

The Disparity Bit ensures deterministically the advantages brought statistically by the scrambling process.

Even with scrambling process, a large running disparity can still occur with very low probability and could produce excessive eye shifts. These eye shifts could be balanced by a more complicated equalization stage in the receiver if the running disparity was still limited. However, a PRBS does not bind the running disparity deterministically, thus the data could be corrupted on the reception end and it could eventually cause the PLL in the CDR to lose its lock. The implementation of the Disparity Bit process prevents from this eventuality.

The transmitter constantly monitors the disparity of the transmission.
For each frame, the running disparity is calculated and 2 cases can occur:

- The running disparity of the transmission **does not** increase above ±64 (+64 and -64 included). In this case, the disparity bit is set to '0' and the 63 bits composed of the scrambled data and of the Clk Bit are transmitted as is.
- The running disparity of the transmission **does** increase above ±64 (+64 and -64 excluded). In this case, the 63 bits composed of the scrambled data and of the Clk Bit are inverted and the disparity bit is set to '1'.

In normal operating mode, the receiver will check the disparity bit first.
If the disparity bit is high then the received data are inverted (including the clk bit), then the data are descrambled.
If the disparity bit is low then data are descrambled only.

1.6 ESIstream 62B64B Synchronization Sequence (ESS)

Each serial lane must be synchronized to align the frames between the transmitter and the receiver and to synchronize the reception scrambler with the transmission scrambler.

The synchronization is controlled through the synchronization signal (SYNC) sent to the receiver and to the transmitter.
The receiver must receive the SYNC signal prior to receive the ESIstream 62B64B Synchronization Sequence (**ESS**).

The transmitter generates the ESS when receiving the SYNC signal. The length of the SYNC pulse should be at least as long as 32 UI.

The ESS is composed of two parts:

- The Frame Alignment Sequence (**FAS**) is composed of 32 frames of a 64-bit programmable COMMA, 0x00FFFF0000FFFF00 or 0xACF0FF00FFFF0000. This sequence bypasses the scrambling, the Clk Bit and the Disparity Bit processing (the sequence is DC balanced). This alignment pattern (COMMA or FLASH pattern) is used by the receiver to align its data on the transmitter output data.

- The PRBS Alignment Sequence (**PAS**), 32 additional frames containing the scrambling PRBS alone. These frames contain 62 bits of the PRBS plus the Clk Bit and the Disparity Bit. These frames go through the disparity processing, as the PRBS value will start to impact the running disparity of the transmission.

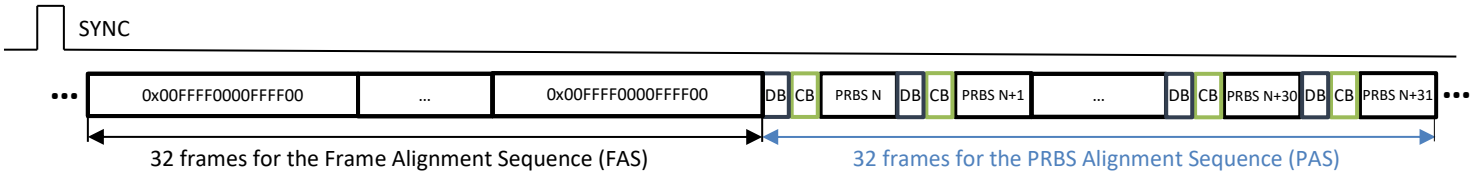

Figure 7: ESIstream Synchronization Sequence (ESS) with COMMA = 0x00FFFF0000FFFF00

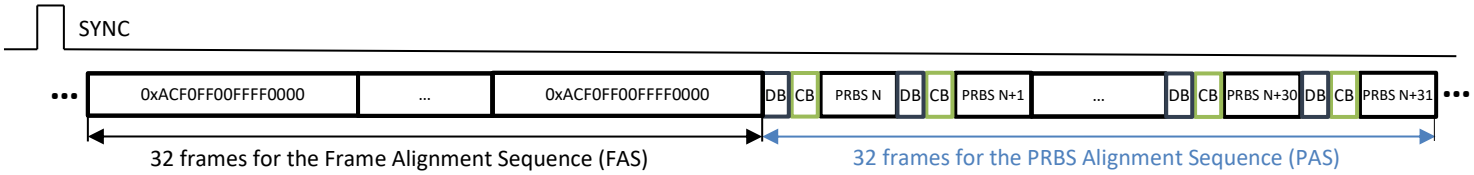

Figure 8: ESIstream Synchronization Sequence (ESS) with COMMA = 0xACF0FF00FFFF0000

When transmitter (**TX**) receives the SYNC, the TX sends the FAS bypassing the scrambling and disparity processing (the sequence is DC balanced). After the FAS, the transmitter sends the PAS.

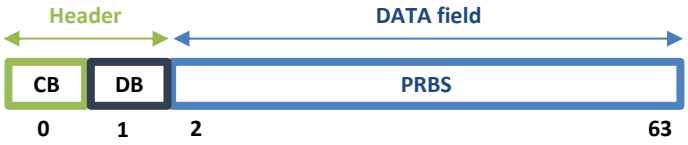

Figure 9: PRBS frames sent during the PRBS Alignment Sequence (PAS)

The receiver will detect the transition from the FAS to the PAS. The PRBS is reset by the transmitter when receiving the SYNC to avoid the first frame of the PRBS initialization being the COMMA; this to ensure that passive detection is precise to the frame.
The receiver will determine its PRBS initial value after receiving 1 valid frame of the PAS. After that, the synchronization of the link is complete.