Encoding Specification

1. Introduction

1.1. Basic Operation

PCIe gen 1 uses scrambling and 8b10b encoding of outgoing data packets to improve the reliability of the electrical communication channel. The receiver reverses this encoding procedure to recover the original data.

1.2. Interaction with Other Modules

The encoding module receives the final outgoing packets from the original PCIe IP and performs scrambling and 8b10b encoding on them. It then sends the resulting data to the PIPE interface instead of the original data. The PIPE interface has to be modified to account for the larger size of the encoded data (each 8 bits are converted into 10 bits). 2 copies of the encoder module are implemented in the Top Transmitter module, 1 for each outgoing byte packet.

2. I/O Signals Description

2.1. Inputs

Name	Width	Source	Description
i_clk	1 bit	**Global**	**Clock signal at positive edge registers**
I_clkx8	1 bit	**Global**	**Another clock signal with 8 times the rate of the main clock.**
i_reset_n	1 bit	**Global**	**Synchronous negative-edge reset**
i_DKin	1 bit	**Input from the Framing_Allignment module**	If it equals 1, the input is a control signal. If it equals 0, the input is a data signal.
i_datain	8 bit	**Input from the Framing_Allignment module**	Incoming data bits

2.2. Outputs

Name	Width	Destination	Description
o_dataout	10 bits	**Output to the modified PIPE interface**	Outgoing encoded data.

3. Detailed Operation

The function of the encoder can be described in the following points:

- The encoder consists of 2 modules, a scrambler and an 8b10b encoding module.
- The scrambler feeds its output to the 8b10b encoding module.
    - The scrambler consists of an LFSR that generates a random sequence.
    - The random sequence is XOR'd with the input data to generate the scrambled output.
    - Control signals are not scrambled and remain the same, only data signals are scrambled.
    - If a COM signal is received, the random sequence of the LFSR is reset (this is used to synchronize the scramblers in the transmitter and the receiver)
- The 8b10b encoding module takes the scrambled data and generates the final output.
    - The 8b10b encoding module functions as a lookup table that takes each input into its corresponding output.
    - Each possible input has only ONE or TWO possible outputs, based on the value of the current running disparity (The difference between the number of outgoing 1 and 0 bits)
    - The 8b10b encoder balances the disparity to maintain DC balance of the electrical network.
    - The D/K signal is also used in determining the output (control and data signals with the same i_datain content will still have different output).
- To reverse the encoding process, the encoded data will have to pass through an 8b10b decoder module by using a reverse lookup table to retrieve the original scrambled data. The scrambled data is then passed through another scrambler that is synchronized with the one in the transmitter (using the COM signal reset) and the output will be the original data.

4. Configuration

4.1. Configuration Sequence

To initialize this module:

- A reset signal will first have to be sent to initialize the entire PCIe device.
- A COM signal will have to be transmitted from the LTSSM to synchronize the scramblers in the transmitter and the reciever.

4.2. Latency

The encoded data bits take **Two** clock cycle to come out.

4.3. Input Format and Constraints

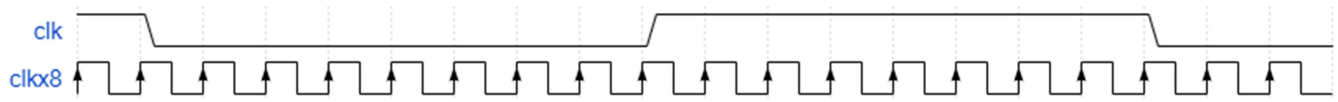
- Only valid control signals are encoded by this module. But any data signal is allowed

5. Example Sequences and Waveforms

5.1. Clkx8:

The clockx8 signal is a secondary clock signal that is in-phase with the main clock but has 8 times the frequency. This extra signal is needed by the scrambler to generate its random numbers.

The waveforms below show the relation between the main clk and the clkx8 signal.



5.2. Reset sequence:

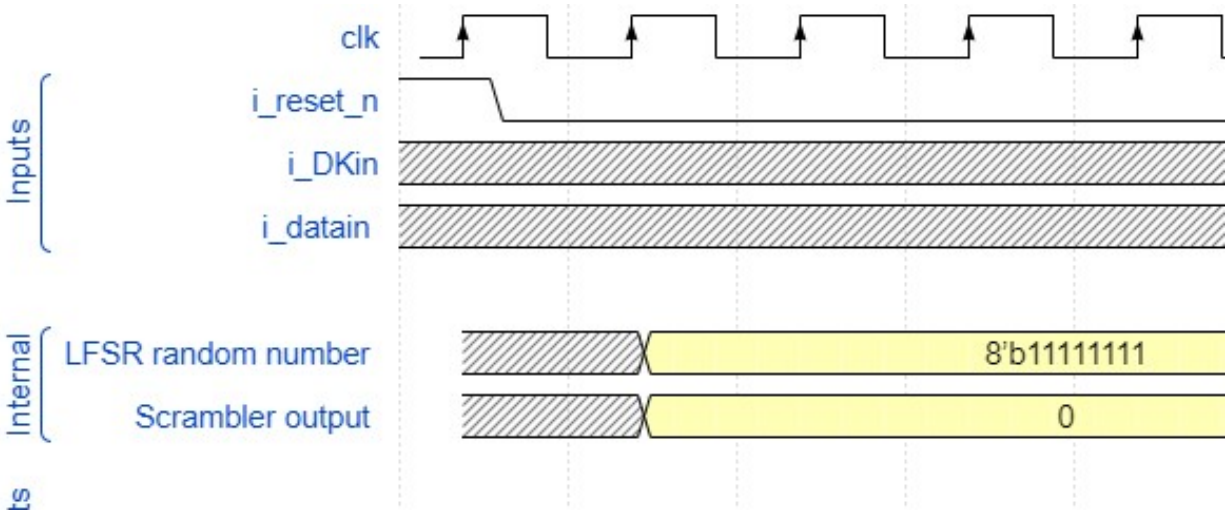
Description:

This sequence resets the state of the encoder.

Signals:

Signal type:	Signal name:	Value
Input	i_reset_n	0
Input	i_DKin	X
Input	i_datain	X
Internal	LFSR random number	8'b11111111
Internal	Scrambler output	0
Output	o_dataout (8b10b encoded data)	0

Waveform(s):



5.3. Normal-Operation sequence:

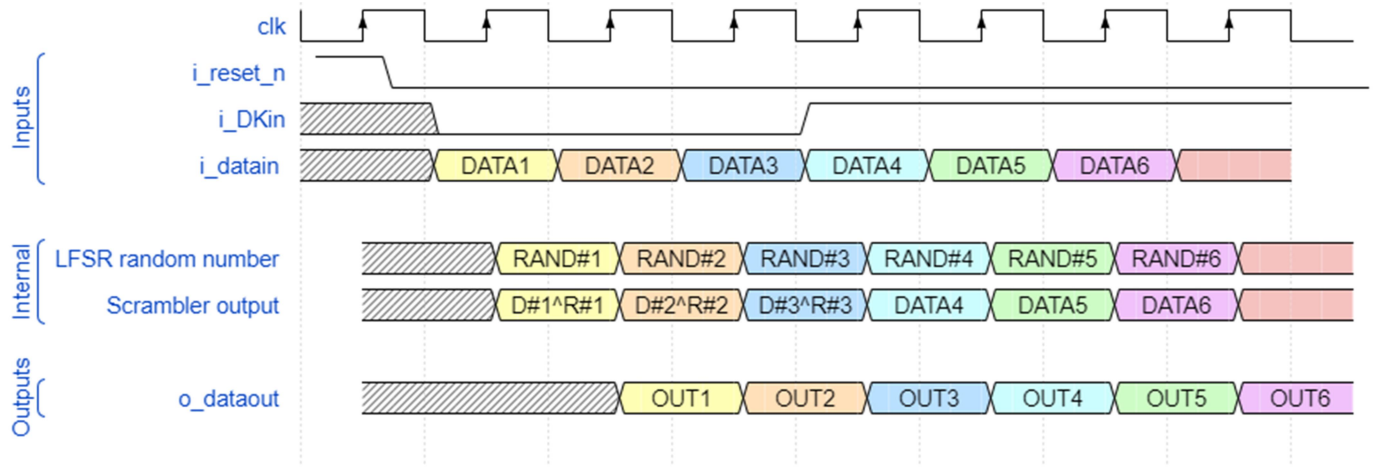
Description:

This sequences showcases the normal operation of the encoder.

Signals:

Signal type:	Signal name:	Value1	Value2	Value3	Value4	Value5	Value6
Input	i_reset_n	1	1	1	1	1	1
Input	i_DKin	0	0	0	1	1	1
Input	i_datain	DATA1	DATA2	DATA3	DATA4	DATA5	DATA6
Internal	LFSR random number	RAND#1	RAND#2	RAND#3	RAND#4	RAND#5	RAND#6
Internal	Scrambler output	DATA1^ RAND#1	DATA2^ RAND#2	DATA3^ RAND#3	DATA4	DATA5	DATA6
Output	o_dataout (8b10-bencoded data)	X	OUT1	OUT2	OUT3	OUT4	OUT5

Waveform(s):



5.4. COM symbol reset sequence:

Description:

When a COM signal (DK=1 and data_in = 8'b10111100), the LFSR of the scrambler is reset and the random number series returns to the initial value. This procedure is used to synchronize the scramblers in the transmitter and the receiver.

Signals:

Signal type:	Signal name:	Value1	Value2	Value3	Value4	Value5	Value6
Input	i_reset_n	1	1	1	1	1	1
Input	i_DKin	0	0	1	0	0	0
Input	i_datain	DATA1	DATA2	COM (8'b10111100)	DATA4	DATA5	DATA6
Internal	LFSR random number	RAND#1	RAND#2	RAND#1	RAND#2	RAND#3	RAND#4
Internal	Scrambler output	DATA1^ RAND#1	DATA2^ RAND#2	DATA3^ RAND#1	DATA4^ RAND#2	DATA5^ RAND#3	DATA6^ RAND#4
Output	o_dataout (8b10-bencoded data)	X	OUT1	OUT2	OUT3	OUT4	OUT5

Waveform(s):