100BASE-T1L PCS Transmit Scrambler

Tingting Zhang, Yan Zhuang

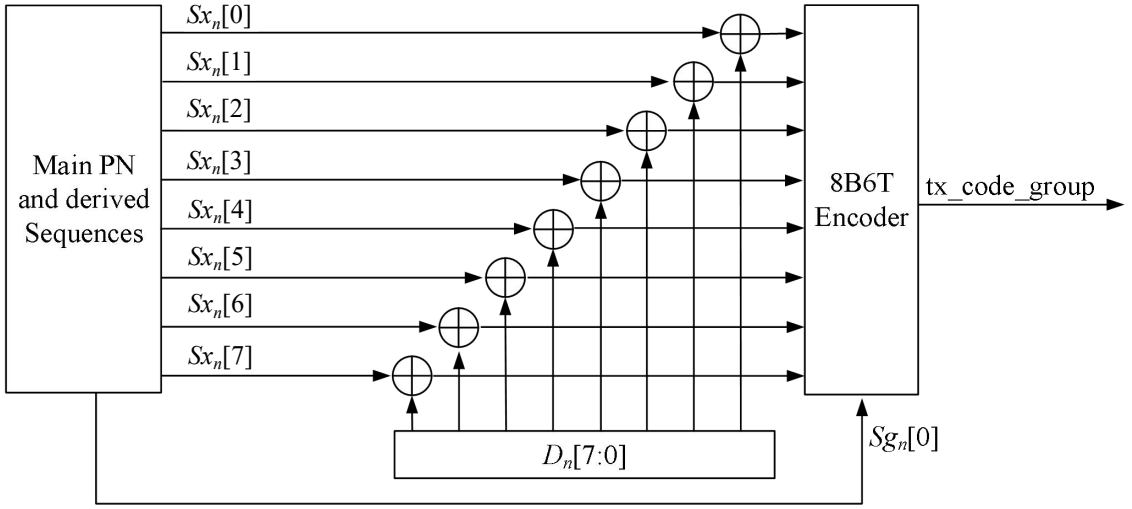
Huawei Technologies

Introduction

- PHY baseline has been achieved up to now:
  - 8B6T PAM3 for data mode
  - Dual-mode PCS:
    - long-reach mode: 64B/65B + 1bit insertion + RS(128,122) + side-stream scrambling + 8B6T
    - low-latency mode: 16B/17B + 1bit insertion + side-stream scrambling + 8B6T
- To do list:
  - the inserted 1 bit TBD (for vendor-specific communication is more favored)
  - data-mode scrambling
  - PMA training
- This presentation discusses data-mode scrambling.

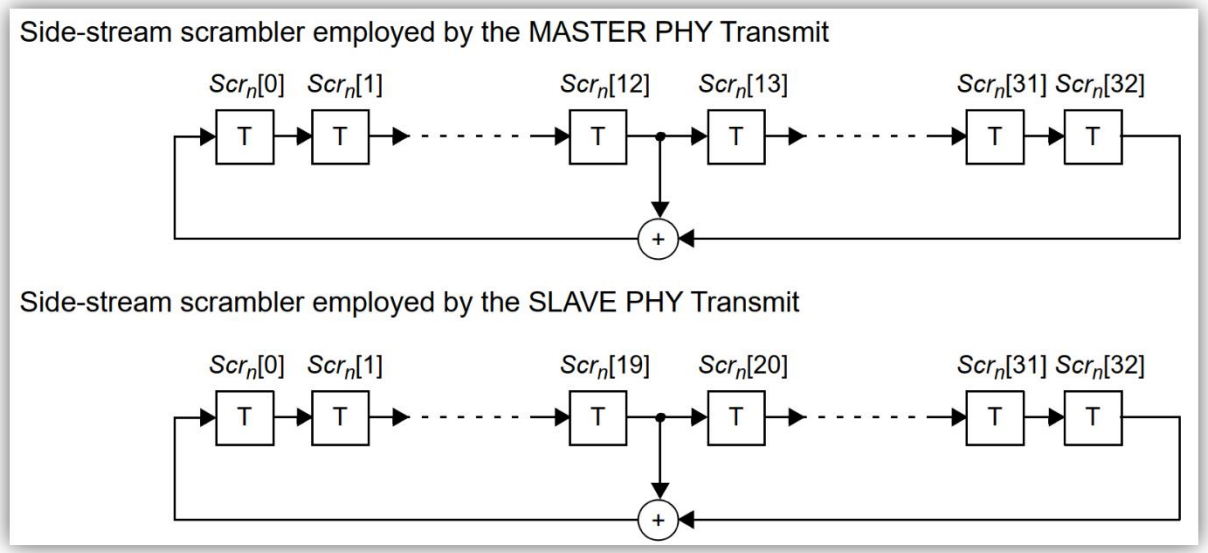
PCS Transmit Scrambler (data mode)

- The PCS Transmit function employs 8 scrambler bits $Sx_n[7:0]$ to decorrelate PHY frames for both long-reach and low-latency modes.
  - The PHY frame consists of either one 1024-bit FEC frame (long reach) or 15 17B blocks and 1 inserted bit (low latency).
- The PCS scrambler may also derive the auxiliary encoding bit $Sg_n[0]$, which decides whether to negate PAM3 6-tuple with positive disparity change when RD=0. Bit $Sg_n[0]$ does not impact interoperability.
- To reduce hardware complexity and improve training efficiency, it is better to use the same side-steam scrambler for both data and PMA training.



Scrambler Bits $Sx_n[3:0]$

- Scrambler bits $Sx_n[3:0]$ are generated based on the bit $Scr_n[0]$ and an auxiliary generating polynomial $g(x) = x^3 \oplus x^8$.
  - Bit $Scr_n[0]$ for Master is generated with the polynomial $g_{\mathrm{M}}(x) = 1 + x^{13} + x^{33}$, while Slave uses $g_{\mathrm{S}}(x) = 1 + x^{20} + x^{33}$.
  - $Sx_n[0] = Scr_n[0]$
  - $Sx_n[1] = g(Scr_n[0]) = Scr_n[3] \oplus Scr_n[8]$
  - $Sx_n[2] = g^2(Scr_n[0]) = Scr_n[6] \oplus Scr_n[16]$
  - $Sx_n[3] = g^3(Scr_n[0]) = Scr_n[9] \oplus Scr_n[14] \oplus Scr_n[19] \oplus Scr_n[24]$



Scrambler Bits $Sx_n[7:4]$

- The four bits $Sx_n[7:4]$ are generated based on the bit X_n and the auxiliary generating polynomial $g(x)$ as:
  - $Sx_n[4] = X_n = Scr_n[4] \oplus Scr_n[6]$
  - $Sx_n[5] = g(X_n) = Scr_n[7] \oplus Scr_n[9] \oplus Scr_n[12] \oplus Scr_n[14]$
  - $Sx_n[6] = g^2(X_n) = Scr_n[10] \oplus Scr_n[12] \oplus Scr_n[20] \oplus Scr_n[22]$
  - $Sx_n[7] = g^3(X_n) = Scr_n[13] \oplus Scr_n[15] \oplus Scr_n[18] \oplus Scr_n[20] \oplus Scr_n[23] \oplus Scr_n[25] \oplus Scr_n[28] \oplus Scr_n[30]$
- The auxiliary encoding bit $Sg_n[0]$ can be generated as:

$$Sg_n[0] = g^4(Scr_n[0]) = Scr_n[10] \oplus Scr_n[12].$$

- Bits $Sx_n[7:0]$ and $Sg_n[0]$ have the same maximum-length shift register sequence of length 2^{33}-1. The associated delays are all large and different so that there is no short-term correlation among these bits.

Receiver descrambler polynomials

- The PCS descrambles the data stream and returns the proper sequence of symbols to the decoding process for generation of RXD<3:0> to the MII.

- Descrambler synchronization is achieved during PMA training.

- For side-stream descrambling, the MASTER PHY shall employ the receiver descrambler generator polynomial $g_{\mathrm{S}}(x) = 1 + x^{20} + x^{33}$. The SLAVE PHY shall employ the receiver descrambler generator polynomial $g_{\mathrm{M}}(x) = 1 + x^{13} + x^{33}$.

Conclusion

- PCS Transmit function employs 33-bit side-stream scrambling for both long-reach and low-latency modes.

- The 8 scrambler bits applied to data octet can be generated in a systematic fashion using two bits ($Scr_n[0]$, X_n) and an auxiliary generating polynomial $g(x)$.

Q & A