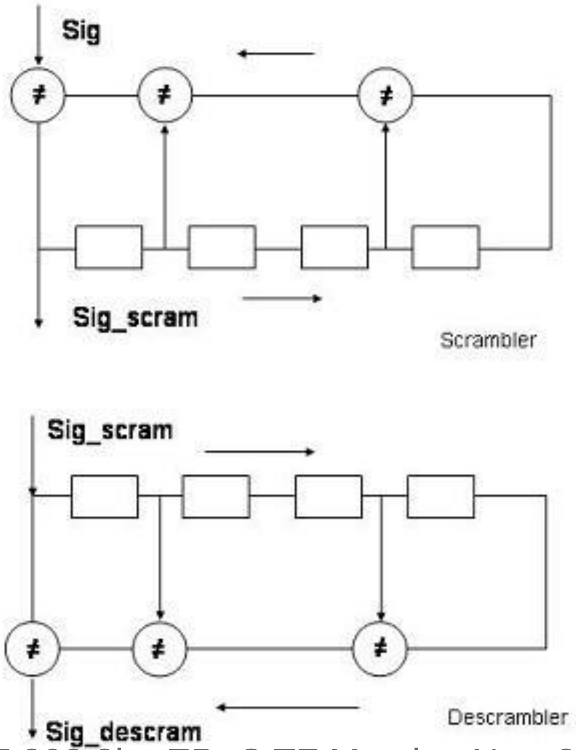
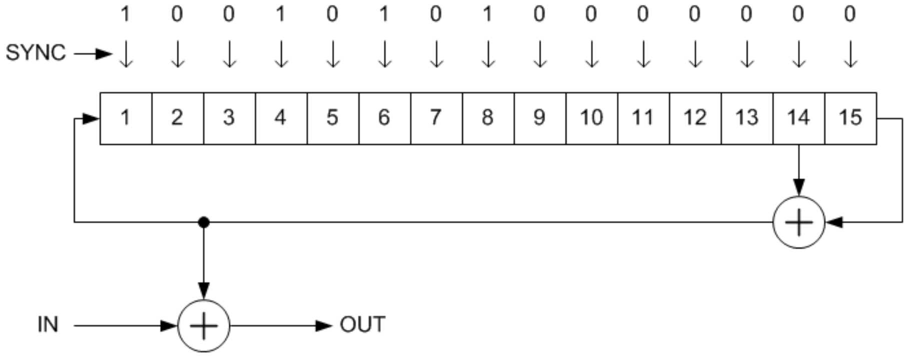
EPoC Scrambler

Jin Zhang (Marvell)

Scrambler Overview

- Multiplicative (self-synchronizing) scrambler
- Additive (synchronous) scrambler





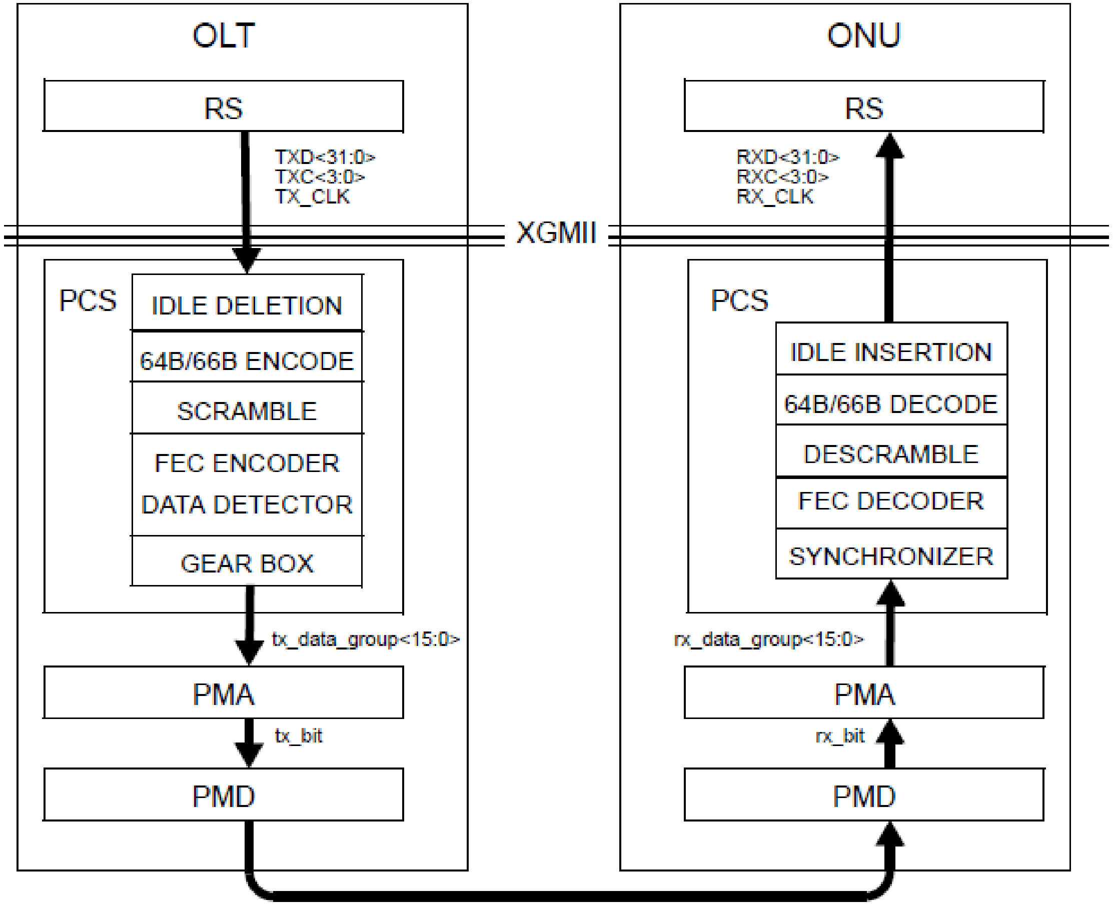
Scrambler Overview

- Self-synchronizing scrambler
  - No need to load the same seed at the receiver
  - Usually at the Tx before FEC, because it will propagate one error into several errors.
  - Can be very long, better randomization and DC balance. For example, in EPON, $g(x) = x^{58}+x^{39}+1$.
  - Data bit loss during synchronization.
- Synchronous scrambler
  - Set/reset seed to synchronize the state at certain point.
  - The effective length of the random sequence of an additive scrambler is limited by the frame length, which is normally much shorter than the period of the PRBS

Requirement of Scrambler for EPoC

- OFDM system does not have DC wander.
- Long runs of zeros and ones cause many carriers to map to same symbol in the constellation, then generate peaks after inverse FFT.
- PAPR or clipping rate in PMD output.

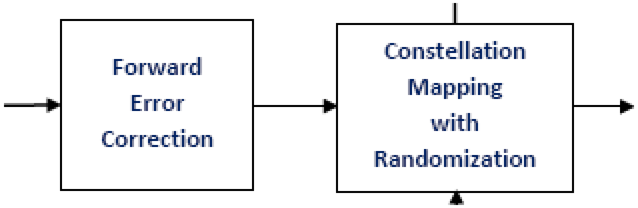
EPON Scrambler



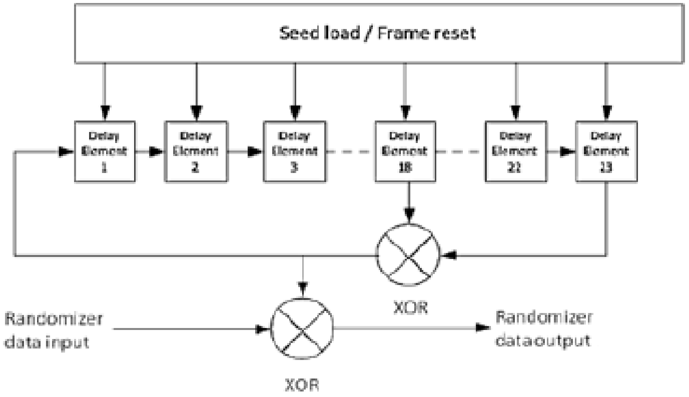
$g(x) = x^{58}+x^{39}+1$

DOCSIS 3.1 DS Randomizer

- Randomize the cell words before constellation mapping to symbols and after the FEC.
- GF(2^{12}) $g(x) = x^2+x+\alpha^{11}$. Equivalent period 2^24-1
- Synchronize to the PLC frame, 128 OFDM symbols.



DOCSIS 3.1 US Randomizer



- After the FEC encoder.
- g(x)=x^23+x^18+1, period 2^23-1
- Synchronize to each burst
- Need to use MAC message to assign the seed.

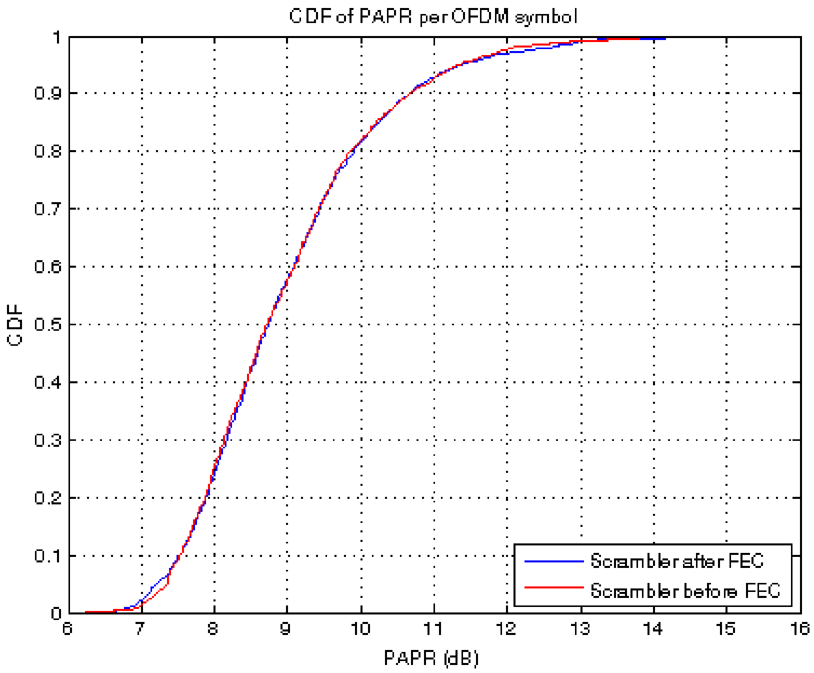
Scrambler and FEC

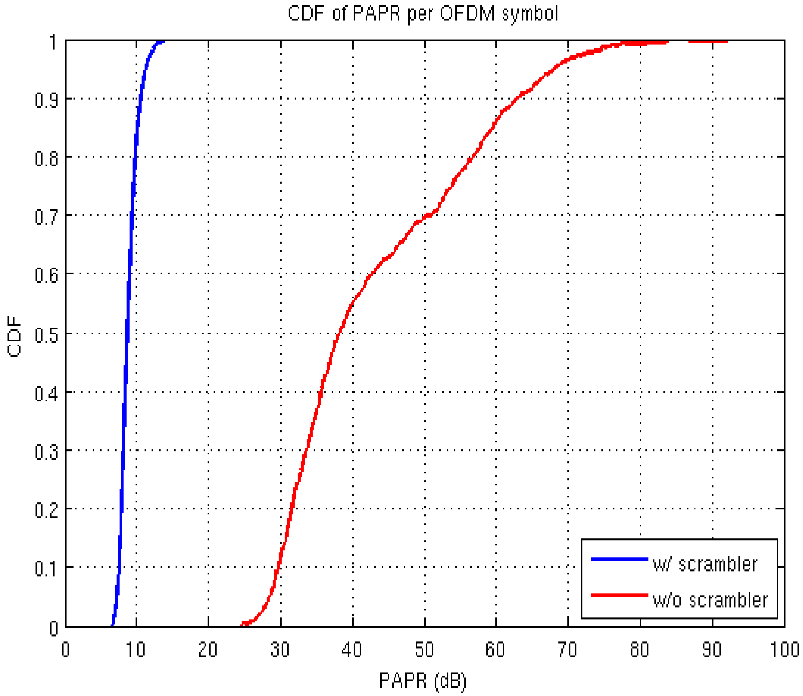
- Usually scrambler is before FEC encoder.
  - Less error propagation for self-synchronizing scrambler
  - The parity of FEC is DC balanced, is usually not scrambled.
  - The interleaver after FEC could play a further role of randomization and reduce the PAPR.
- LDPC has long sets of parity bits, does it need to be scrambled?

Simulation Conditions

- 4096 FFT, 3840 subcarriers, 1024QAM, LDPC (16200, 14400)
- Information bits: 13400, with 1000 zero padding.
- Scrambler before FEC: use EPON self-synchronizing scrambler
- Scrambler after FEC: use D3.1 US scrambler.
- Also show result of no scrambler

Simulation





Add simulation results for comparison with the case of no scrambler

Consideration of Scrambler

- Little difference on PAPR as to scramble or not to scramble the parity of the LDPC codeword.
- Time and frequency interleaver can further randomize the parity bit.
- Complexity wise:
  - Self-synchronizing scrambler, e.g. EPON scrambler, is simpler, stream based processing, no message exchange, no jitter, no uncertainty in latency.
  - Synchronous scrambler, e.g. DOCSIS 3.1 randomizer: need set/reset seed, message for exchange of seed, block based processing, latency is implementation dependent.

Error Propagation and Data Loss for Self-synchronizing scrambler (New Slide)

- Error Propagation
  - 1 bit error -> 3 bit errors
  - (16200, 14400): loss 0.05dB (estimated from [1])
  - (5940, 5040): loss 0.1dB
  - (1120, 840): loss 0.3dB
- Data Loss
  - Every codeword error: loss of 58 bits or 0.4% more data loss, due to one codeword error for length 16200, 1.15% more loss for length 5940, 6.9% for length.

Conclusions (New Slide)

- Scrambler before or after FEC? Hard to find any difference in terms of PAPR loss, as long as it is bit-level scrambling.
- Scrambler after FEC may increase the implementation complexity slightly, cause uncertainly in latency and may need distribution of seed.
- Error propagation causes SNR loss and data loss.
- Since EPoC may be well working on the target FER and FLR, the worst case (for short codeword) SNR loss is not desirable.
- Suggestion: adopt the scramblers specified in D3.1

Reference (New Slide)

- [1] Evaluation of Proposed FEC Codes for EPoC, Rich Prodcan, BZ. Shen, http://www.ieee802.org/3/bn/public/sep13/prodan_3bn_01_0913.pdf