High spectrally efficient coded 16-PAM scheme for GEPOF based on MLCC and BCH

Rubén Pérez-Aranda
(rubenpda@kdpof.com)
David Ortiz
Dunia Prieto

Agenda

- Background and objectives
- Channel model
- Spectral efficiency and baud-rate selection
- Research on coded 16-PAM schemes
- Coded 16-PAM based on MLCC and BCH

Background & Objectives

- In the GEPOF SG was demonstrated the necessity of high spectrally efficient coding schemes to approach the POF channel capacity and therefore to meet the link budget requirements

- The analysis based on Information Theory suggested that the combination of high spectrally efficient coded Pulse Amplitude Modulation (PAM) with Tomlinson-Harashima Precoding (THP) is a feasible solution (see [1])

- Then, several coded modulation schemes were studied in [2] providing a comparison between them in terms of coding gain and complexity and reinforcing the technical feasibility of the approach

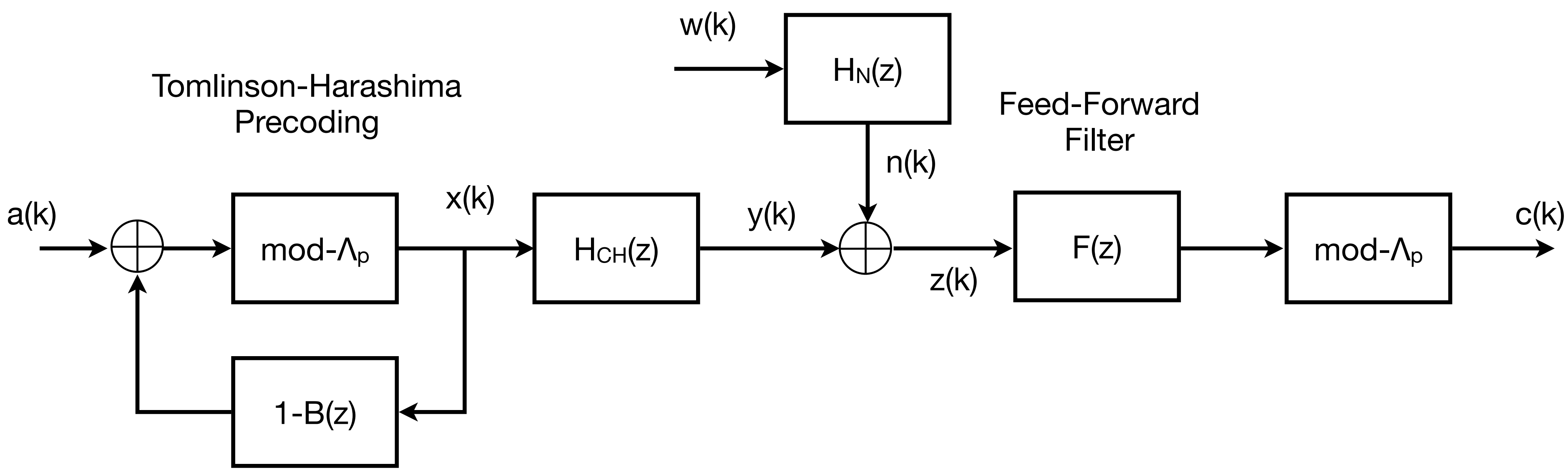
- A deeper research on optimal schemes for GEPOF has been carried out, including in the analysis an important figure of merit: the latency. A report on this research is provided

- Among all the possible schemes, one has been selected. This presentation provides the full definition of the selected coded PAM scheme for GEPOF

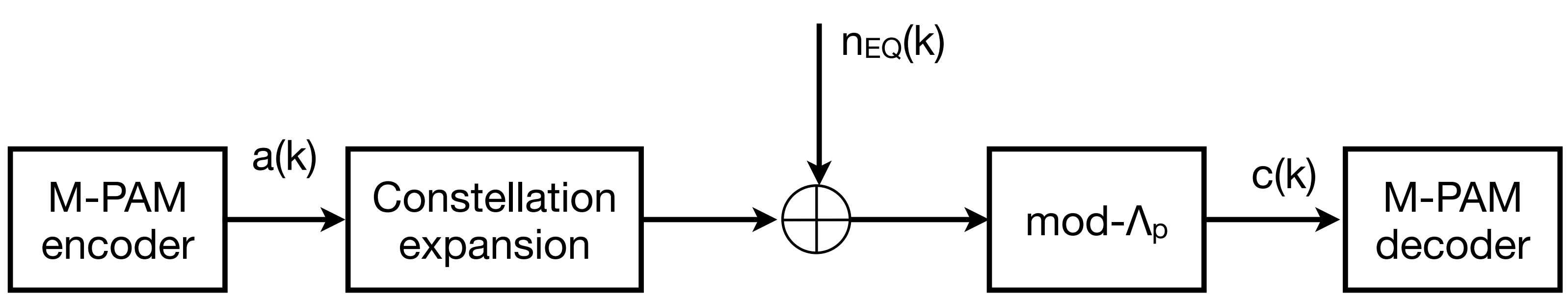
Channel model

Discrete-time equivalent channel model

- Based on [2], an equivalent linear discrete-time channel model can be built by sampling the continuous-time optical channel
- THP at input of the channel and feed-forward equalization at the receiver are added to compensate the ISI and whiten the noise and to take into consideration the capacity penalties introduced by the equalization process



Equivalent after equalization



$$SNR \triangleq \frac{\sigma_a^2}{\sigma_{n_{EQ}}^2}$$

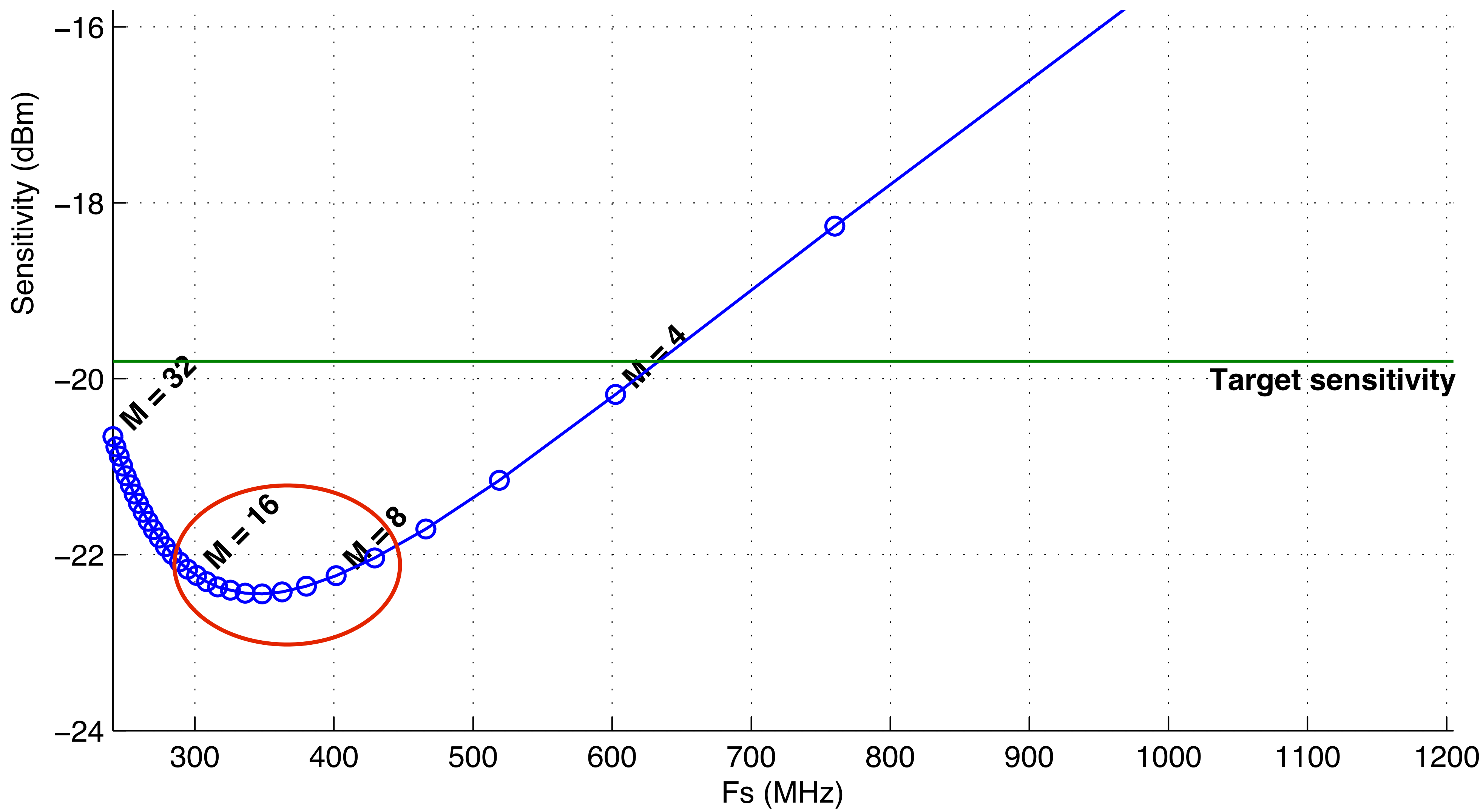
Channel model - general considerations

- A channel model is defined to evaluate the performance of the proposed coded-modulation schemes for GEPOF
- Channel model is defined at the output of the equalizer, according to the following assumptions:
  - ISI has been fully compensated by the THP plus feed-forward equalizer
  - Feed-forward equalizer fully whitens the channel noise
  - Then, the equivalent channel is a memoryless channel with additive white gaussian noise ➤ AWGN channel
  - Therefore, it is assumed neither residual ISI nor colored noise exist
- Modulo operation may be advantageously embedded into the M-PAM decoder to avoid symbol flipping
- The THP power capacity losses due to crest-factor and precoding loss (see [1]) are included in the SNR of equivalent memoryless AWGN channel
- The THP coding penalty due to constellation expansion and modulo operation (noise aliasing) is going to be considered in the evaluation of the coded modulation schemes

Spectral efficiency and baud-rate selection

Spectral efficiency and baud-rate selection

- The Shannon capacity analysis of [1] suggests that the optimal scheme should be a M-PAM with M between 8 and 16, for THP channel and code-rate ~0.8
- The baud-rate has to be between 300 and 400 MHz. In [2] is argued that baud-rates closer to 300 MHz will provide smaller DSP power consumption for channel equalization as well as simpler and lower consumption designs of DAC and ADC

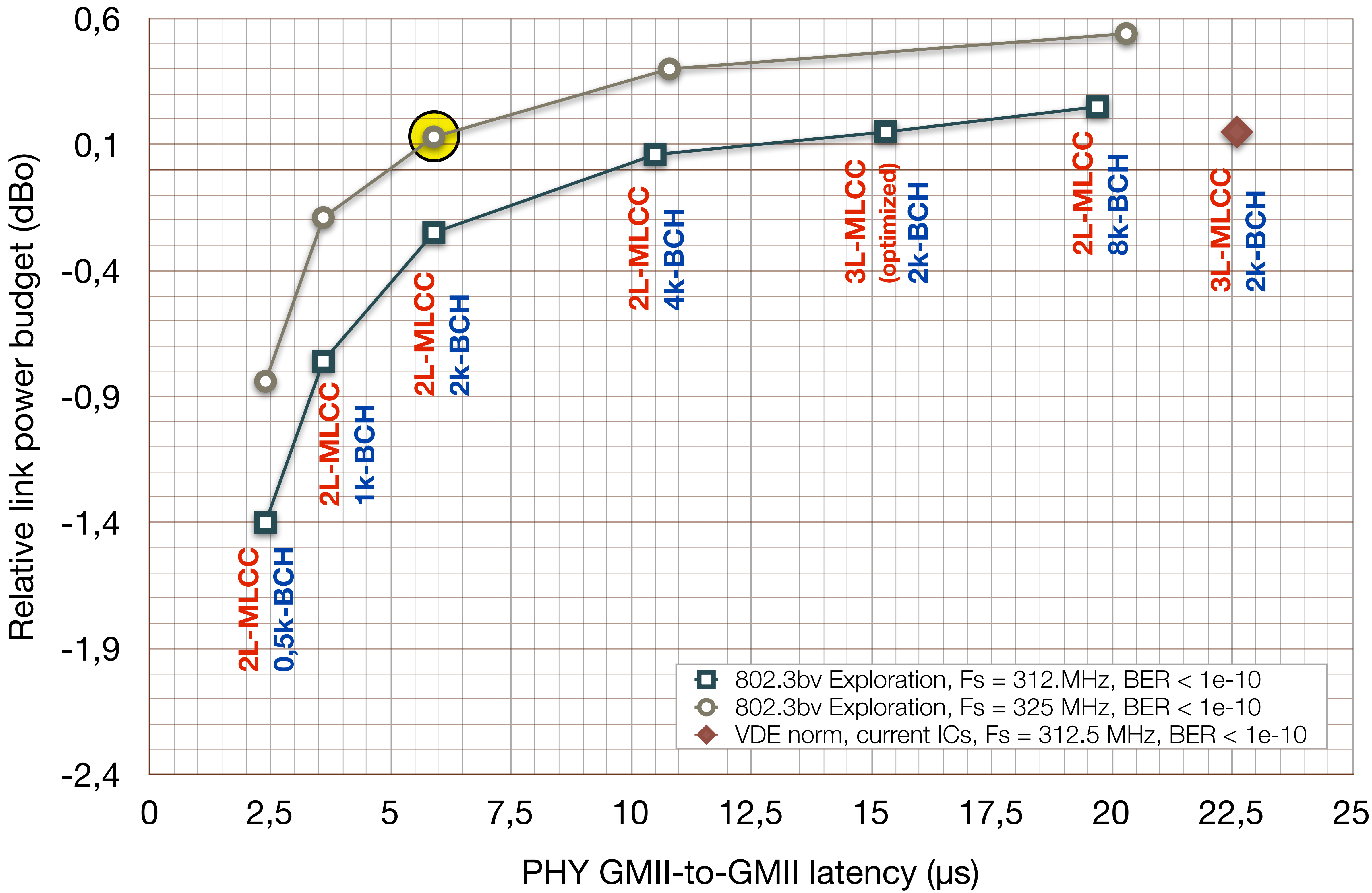


Spectral efficiency and baud-rate selection

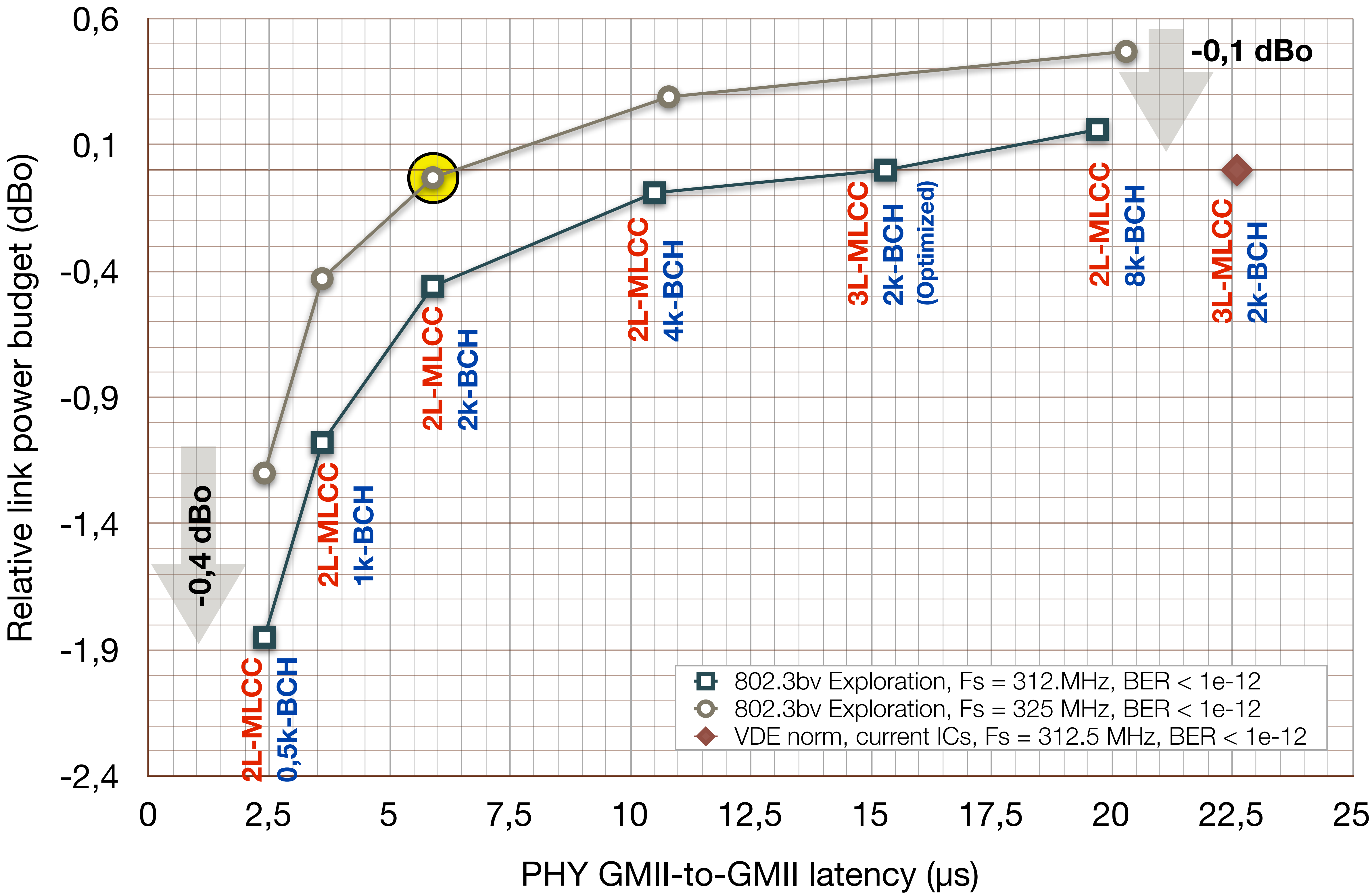
- Selection of the optimal spectral efficiency and baud-rate is a complex task that requires to take in consideration several factors to get a solution:
  - Tradeoff between latency and coding gain
  - Implementation complexity of the FEC and DSP
  - Memory requirements
  - Statistical analysis of how the errors are produced at the output of the FEC
  - Error detection capabilities of the FEC in addition to the correction ones
  - Discrete design space of implementable codes
  - A holistic approach to consider important elements like scramblers, transmission structure, how the timing is recovered, the channel is equalized, etc.
  - Clock references and clock speeds of data interface with MAC (i.e. GMII)

Research on coded 16-PAM schemes

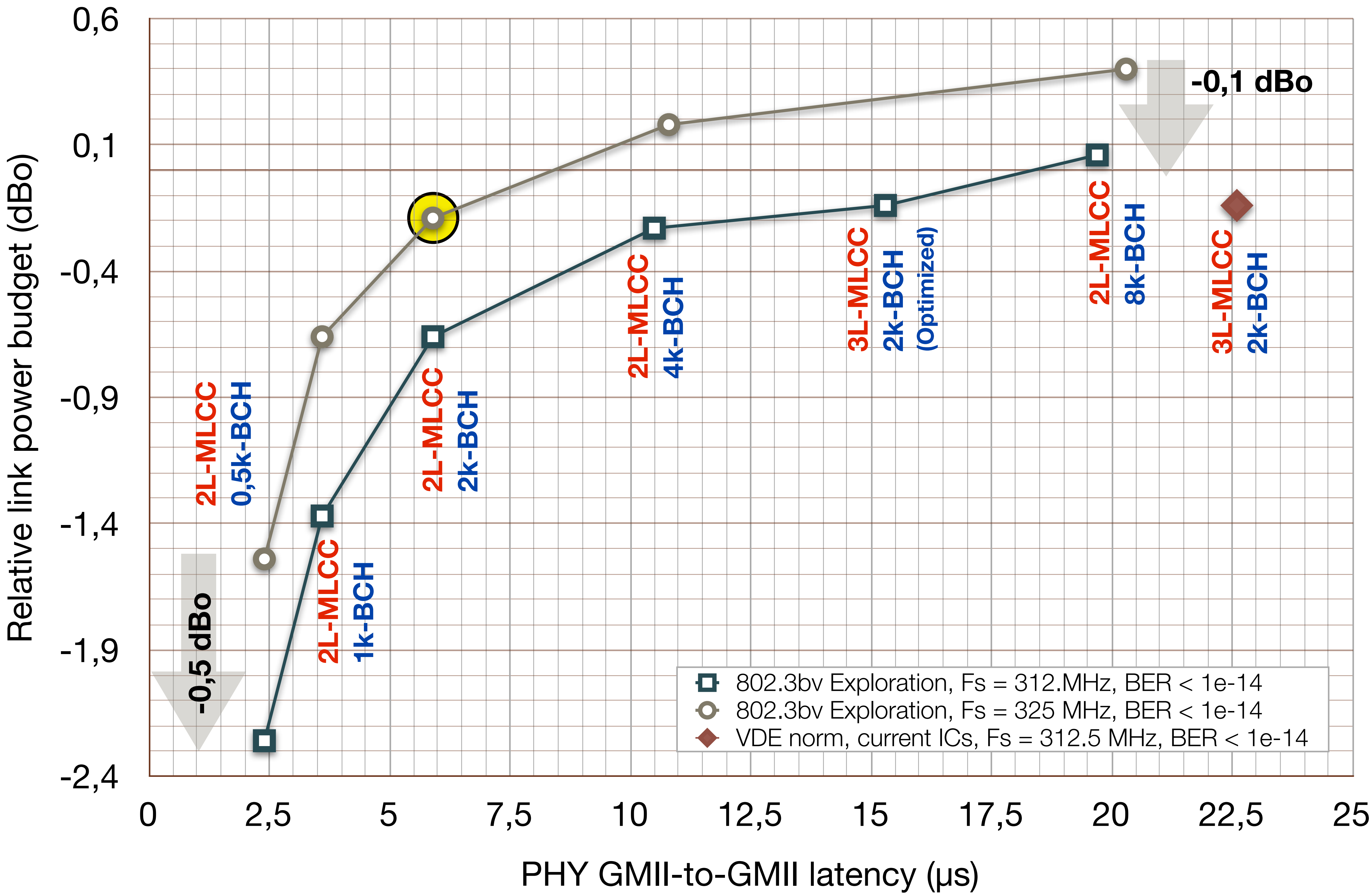
Coded 16-PAM schemes - link budget vs. latency



Coded 16-PAM schemes - link budget vs. latency



Coded 16-PAM schemes - link budget vs. latency



Coded 16-PAM schemes - analysis

- Link budget for low latency FECs is highly affected by the BER specification, because "shorter" codes provides smaller error correction capabilities, being BER more dependent on the SNR of the decoder
- For "longer" codes (high latency), the link budget variation as a function of BER is approximately constant. These codes show a harsh BER vs. SNR performance curve
- The code shown in yellow is a "sweet spot":
  - Low latency: ~6.0 us
  - Same link budget performance of the current ICs implemented according to VDE norm
  - Low implementation complexity and low power consumption
  - Very robust against unconsidered impairments and/or implementation losses
  - Very high MTTFPA
- This code has been selected as optimal solution for GEPOF and full definition of it is provided in following slides

Coded 16-PAM based on MLCC and BCH

Coded 16-PAM - introduction

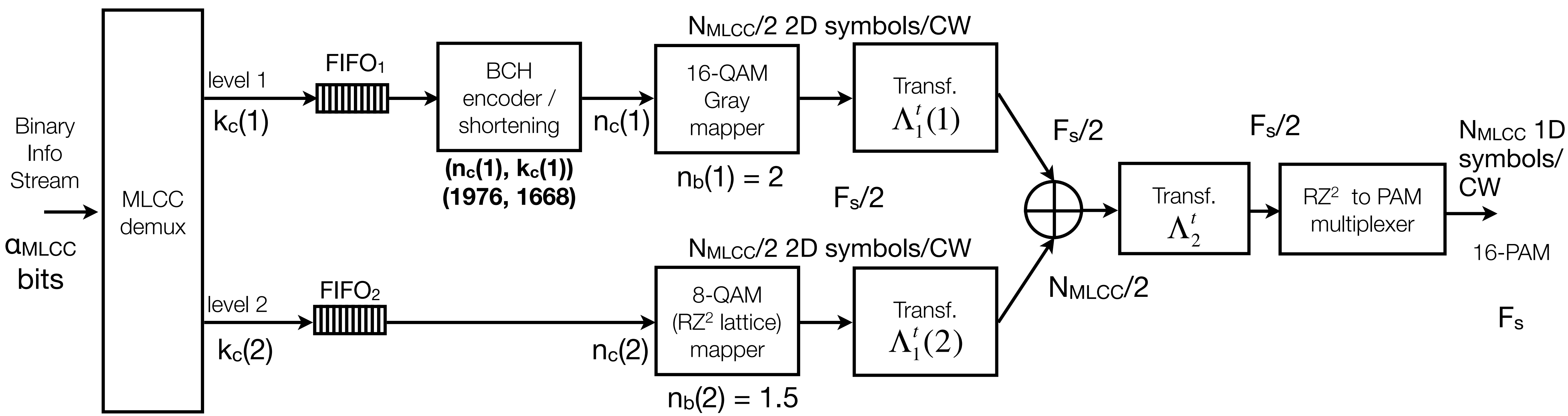
- Multilevel Coset Coding (MLCC) of 2 levels based on Z^2 and RZ^2 lattices to adjust accurately the spectral efficiency with low complexity binary component codes
- The constellation is partitioned in such a way the bits more likely to be corrupted by noise are protected by a binary code, and those bits less corrupted are no protected
- Coded-modulation based on the theory published in [3]
- Theory behind this code is similar to that used for 10GBASE-T FEC

Coded 16-PAM - notation

- Definitions:
  - N_{MLCC}: length of the MLCC code-word in 1D (PAM) symbols
  - $n_b(i)$: nº of coded bits per dimension for the i^{th} level; it defines constellation. i=1..2
  - $n_c(i)$: nº of bits per code-word for the i^{th} binary component code
  - $k_c(i)$: nº of information bits per code-word for the i^{th} binary component code, equal to the nº of information bits per MLCC code-word per i^{th} level
  - ξ: total number of coded bits per dimension: $\xi = \sum_{i=1}^{2} n_b(i)$
  - k_{PAM}: nº of bits per PAM constellation at the encoder output: $k_{PAM} = \lceil \xi \rceil$
  - α_{MLCC}: nº of information bits per MLCC code-word
  - $r_c(i)$: code-rate for the i^{th} level: $r_c(i) = k_c(i) / n_c(i)$
  - η: spectral efficiency per dimension: $\eta = \sum_{i=1}^{2} n_b(i) r_c(i)$
  - F_S: symbol frequency (baud-rate)

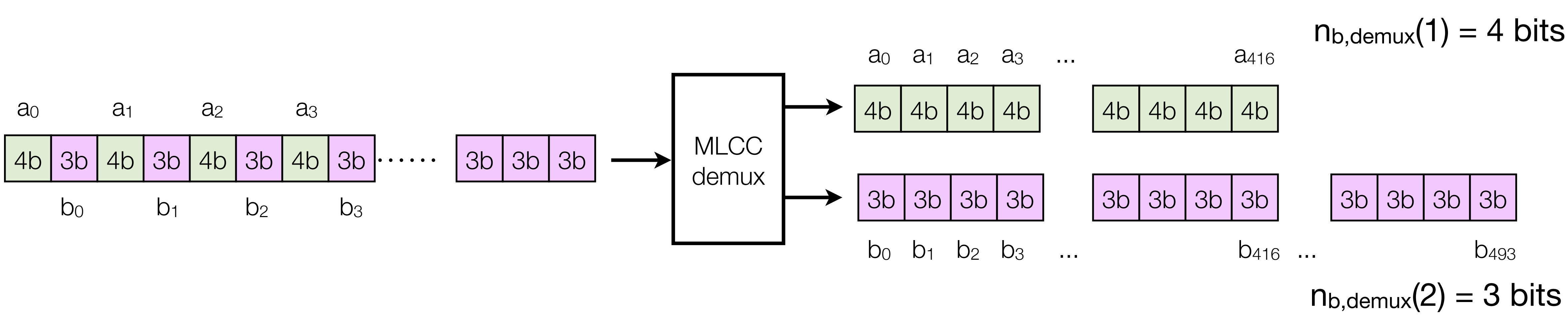
Coded 16-PAM - encoder

- Encoder scheme:
  - **2 levels** MLCC based on BCH component codes, providing 16-PAM in the output
  - Block oriented encoding: 1 code-word is composed by N_{MLCC} PAM symbols
  - N_{MLCC} = 988 symbols
  - BCH code is defined in $GF(2^{11})$
  - α_{MLCC} = 3150 bits, $n_c(1)$ = 1976, $k_c(1)$ = 1668 bits, $n_c(2) = k_c(2)$ = 1482 bits
  - $n_b(1)$ = 2 bits/dim, $n_b(2)$ = 1.5 bits/dim
  - F_s = 325 MHz, η = 3.1883 bits/s/Hz/dim, ξ = 3.5 bits/dim



Coded 16-PAM - MLCC demultiplexer

- Demultiplexing of input binary stream to split the input information among the different levels that compose the MLCC encoder
- The bit ordering is established to achieve the minimum latency of the encoder as well as the Multi-Stage Decoder (MSD)
- For the i_{th} level (i=1..2), we define $n_{b,demux}(i) = 2 \cdot n_b(i)$ bits
- The input information is split between the two levels by allocating $n_{b,demux}(1)$ bits to the first level and $n_{b,demux}(2)$ bits to the second level cyclically until $k_c(1)$ bits have been assigned to the first level.
- Once the first level is full, the remaining input bits are assigned to the second level until second level is full being achieved $k_c(2)$ bits



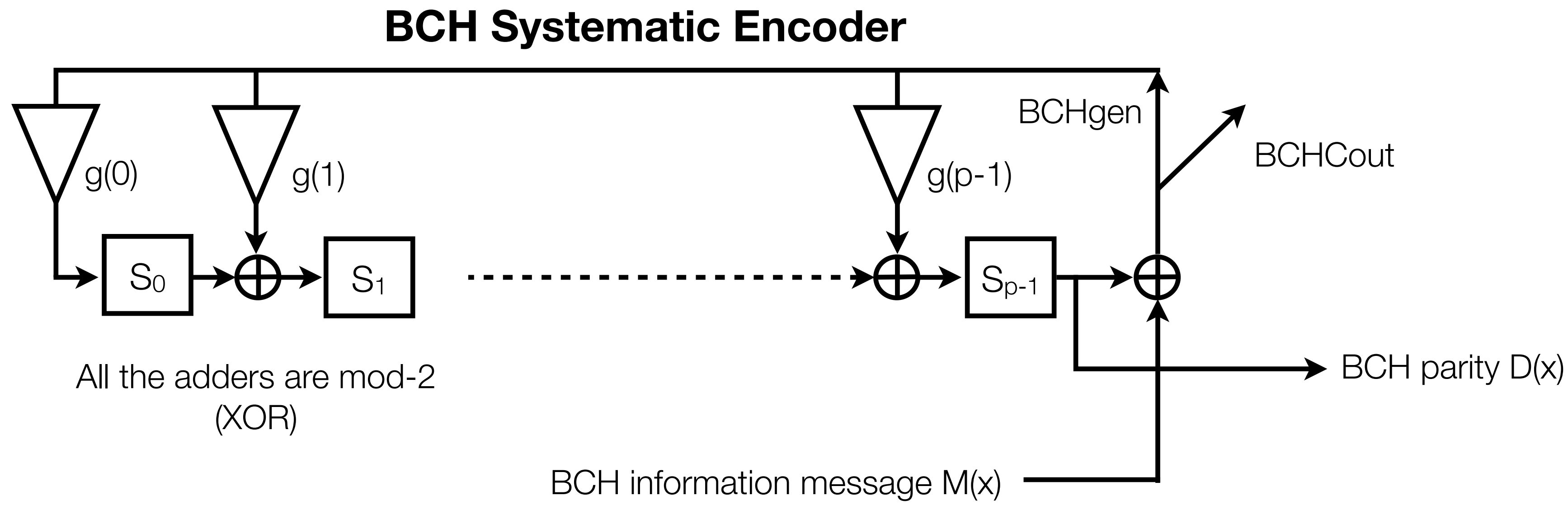
Coded 16-PAM - component code

- This is a shortened version of primitive **BCH (2047, 1739)**
- BCH over Galois' Field GF(2^m), where m = 11 and error correction capability **t = 28**
- **n_c(1) = 1976 bits**
- **k_c(1) = 1668 bits**
- p_c(1): Parity. **p_c(1) = 308 bits**
- r_c(1) = 1668/1976 = ~0.8441
- Shortening is implemented prepending **71** zero bits to **1668** data bits
- In order to minimize the Galois Field Arithmetic we choose as primitive the irreducible polynomial of minimum weight over GF(2^{11}): $1 + x^2 + x^{11}$
- The Generator Polynomial is given by $G(x) = \sum_{i=0}^{p_c} g(i) \cdot x^i$ where *g(i)* takes values 0 or 1
- The order of *G(x)* for this BCH code is **308**
- The G(x) coefficients are given by:

  'h0014_B624_90DF_0781_4D88_99E9_B9DB_6267_00D3_7A90_49DB_C0C4_484A_D6C5_49AB_AE7E_6F58_A406_CF86_C0BD

  being *g(0)* de Least Significant Bit (LSB)

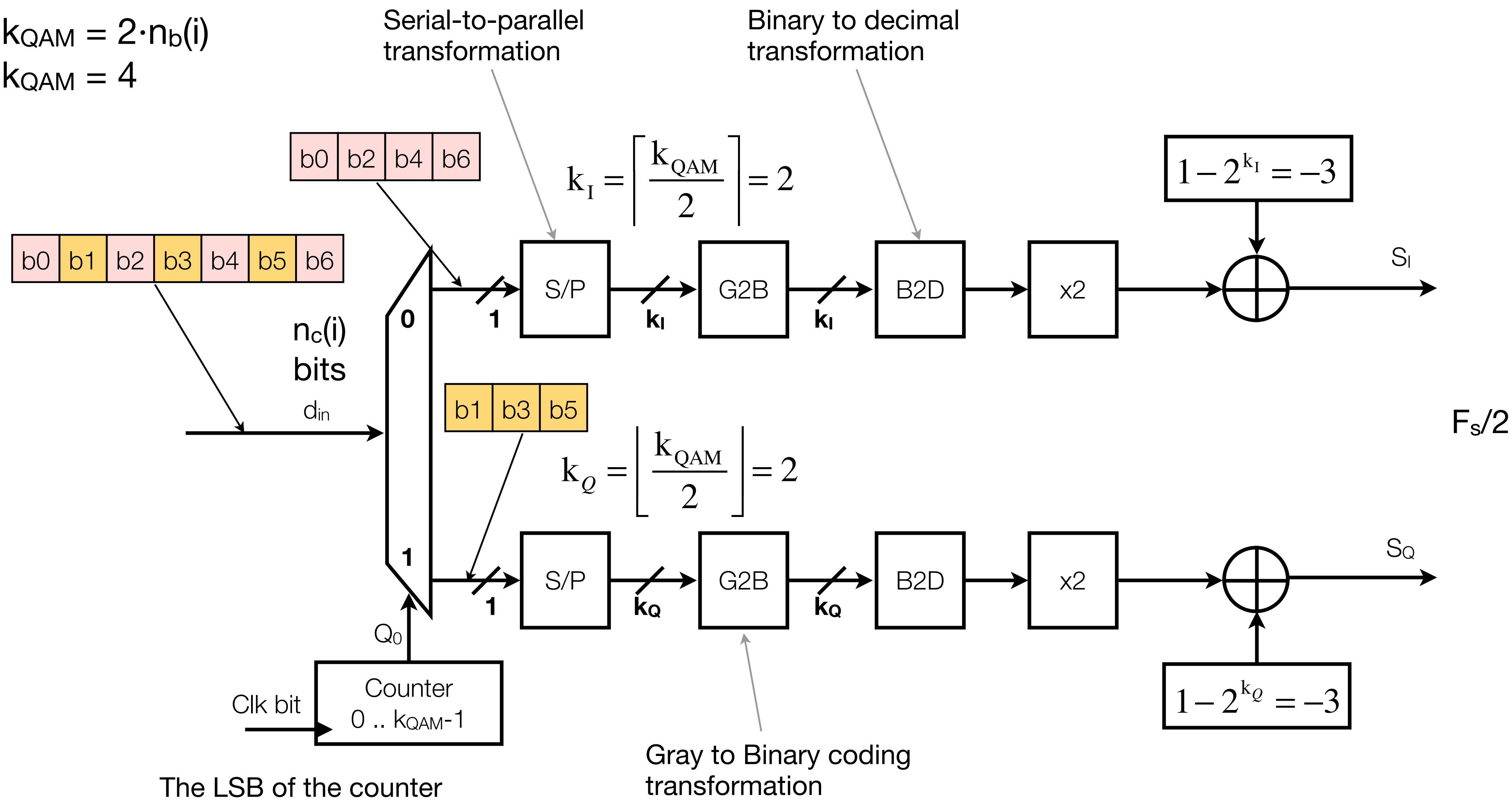
Coded 16-PAM - BCH encoder



- The BCH encoding is systematic and parity is transmitted after the information message
- Encoder needs of two steps to compute parity:
  - Multiply M(x) by x^{n-k}
  - D(x) is defined as the remainder of divide $M(x) \cdot x^{n-k}$ by G(x)
- All the delay elements S_0, ... S_{p-1}, shall be initialized to 0, before encoding
- All the k bits composing the information message M(x) are used to calculate the parity D(x) with switch connected (BCHgen setting).
- After all the k bits have been serially processed, the switch is disconnected (BCHout setting) and the p-1 stored values ($S_0 \ldots S_{p-1}$) are the parity D(x).
- D(x) is transmitted in order from S_{p-1} to S_0.

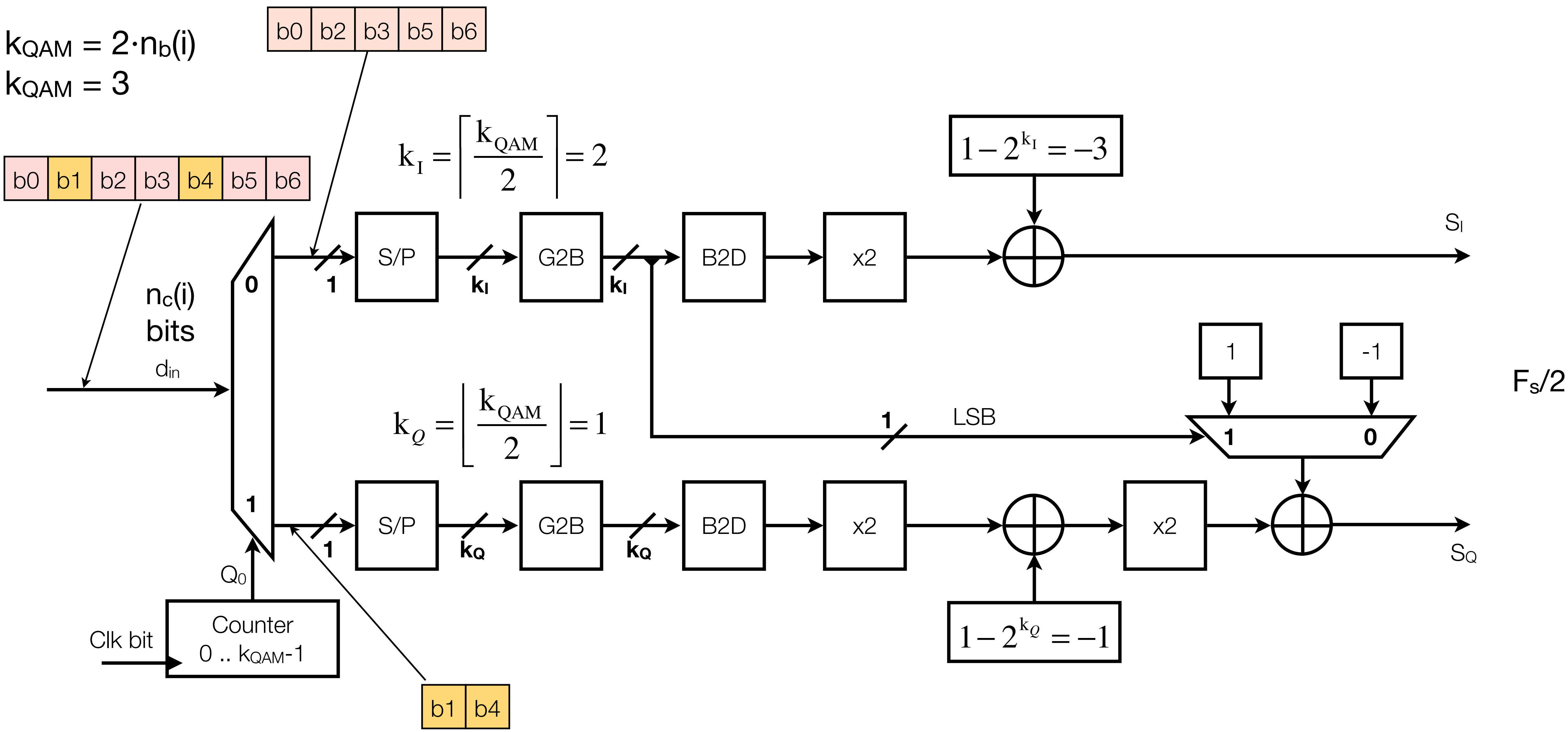
Coded 16-PAM - 16-QAM Gray mapper

$k_{QAM} = 2 \cdot n_b(i)$
$k_{QAM} = 4$



The LSB of the counter output is used to control the demux.
The reset state of counter should zero.
Because the counter is reset for each set of K_{QAM} bits, always start at 0 for each new CW encoding

Coded 16-PAM - 8-QAM quasi-Gray mapper



Coded 16-PAM - mapper operators

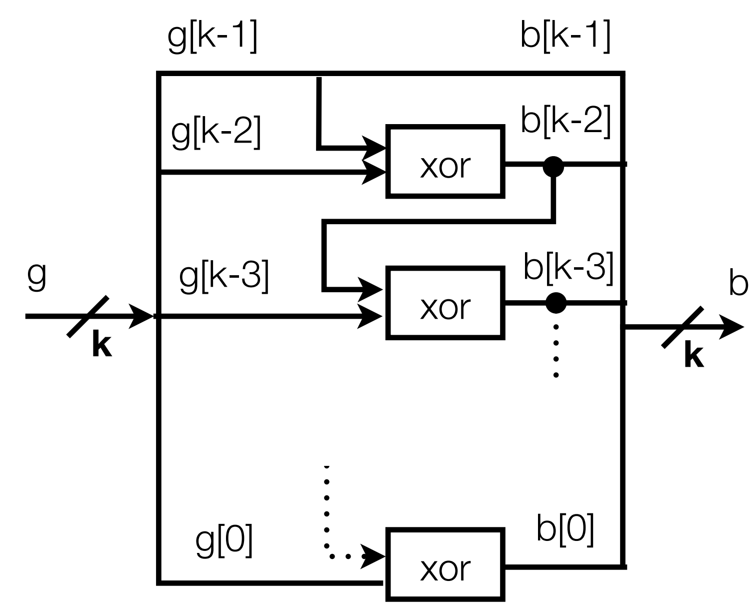
- S/P: serial-to-parallel transformation is performed in such a way the first received bit in the input belonging to a group of k bits is assigned to the LSB of the parallel output, and the last received bit of the group to the MSB

- B2D: unsigned binary to decimal converter. Let be [x(0) .. x(k-1)] the parallel input, where x(i) is the value for each bit of the bus and x(0) is the LSB and x(k-1) is the MSB. The integer output is defined such that

$$d = 2^{x(0)} + 2^{x(1)} + \ldots + 2^{x(k-1)}$$

- G2B: Gray-to-Binary transformation:

$$b[k-1] = g[k-1]$$
$$b[k-1-j] = g[k-1-j] \oplus b[k-j] \quad \forall j \in [1, k-1]$$



Coded 16-PAM - 1st lattice transformation

- Lattice transformation $\Lambda_1^t(l)$
  - Defined for each l[th] level, to accommodate lattices before addition, performing the so called coset partitioning
  - This transformation is composed by 3 operations:
    - Lattice is translated, so that constellation is contained within the first 2D quadrant
    - Lattice is scaled to enable coset partitioning by vectorial addition
    - Lattice is rotated 45º before addition for constellations $\subset nRZ^2$
  - Translation operation:

$$\Lambda_{1,1}^t(l)(x) = x + (1+j)\left(2^{\lceil n_b(l) \rceil} - 1\right) \quad \forall x \in \mathbb{C}, \quad j = \sqrt{-1}$$

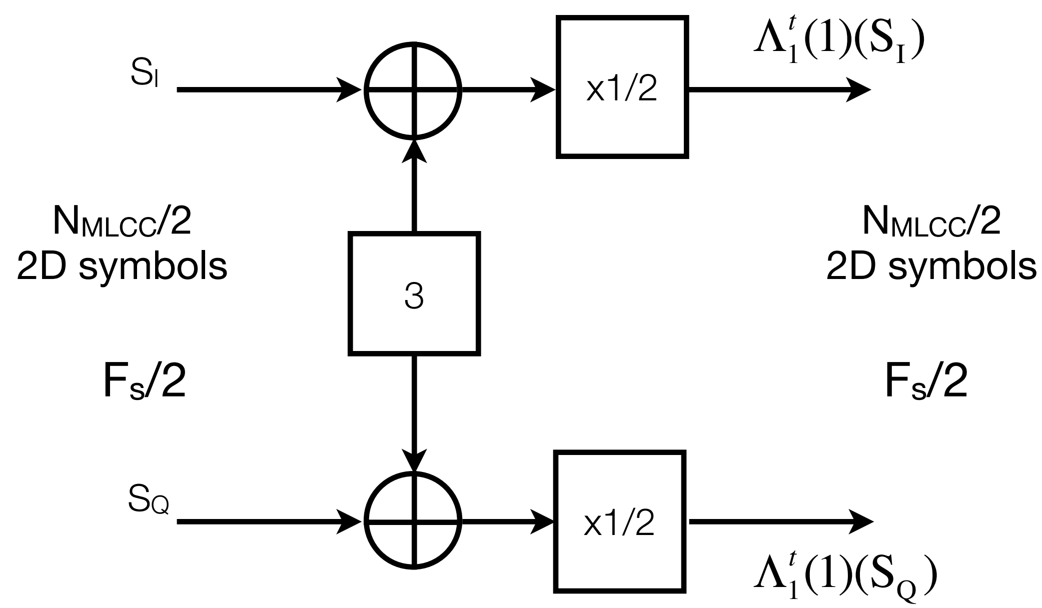
  - Scaling and rotation operations:

$$\Lambda_{1,2}^t(l)(x) = \frac{1}{2} x \cdot 2^{\sum_{i=1}^{l-1} \lceil n_b(i) \rceil} \cdot \left(\frac{1+j}{2}\right)^{\mathrm{rem}(2 n_b(l), 2)} \qquad \forall x \in \mathbb{C}$$

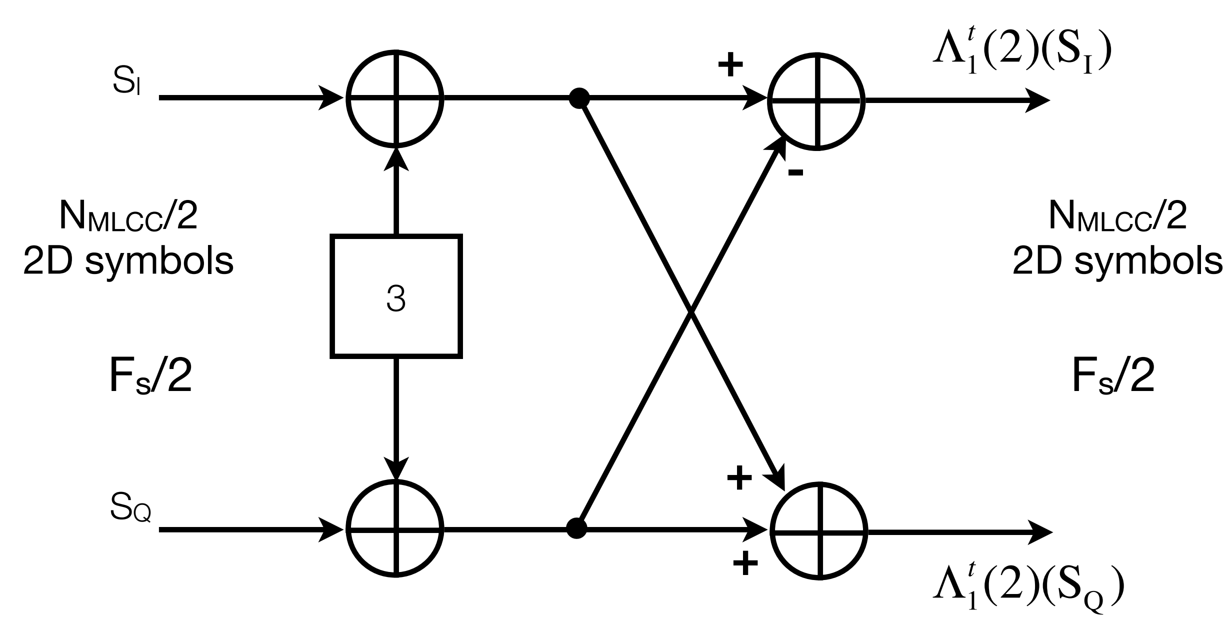
  - Whole transformation is defined as: $\Lambda_1^t(l)(x) = \Lambda_{1,2}^t(l)\left(\Lambda_{1,1}^t(l)(x)\right)$

Coded 16-PAM - 1st lattice transformation

- Lattice transformation $\Lambda_1^t(1)$

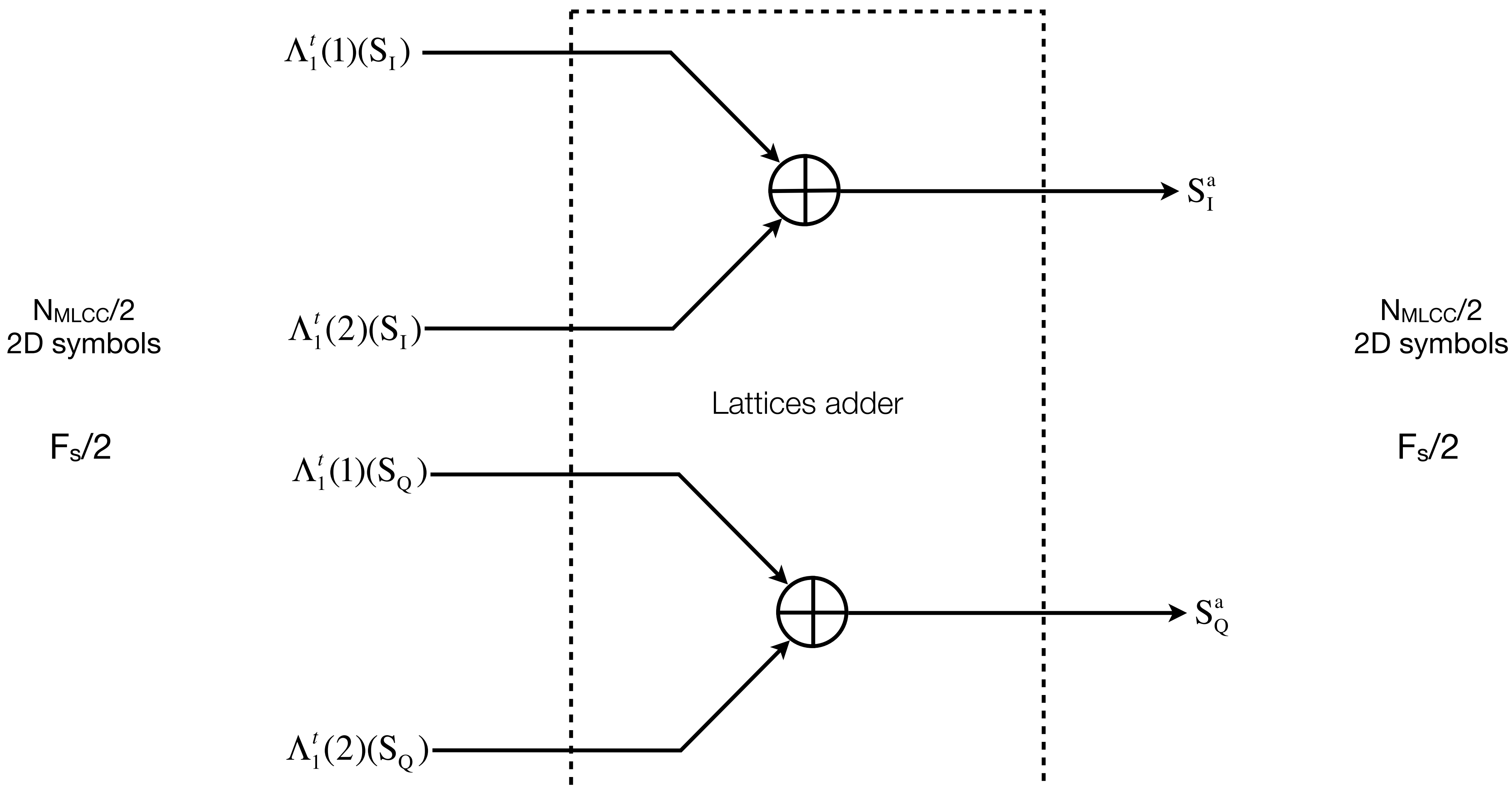


- Lattice transformation $\Lambda_1^t(2)$



Coded 16-PAM - lattices addition

- Lattice addition
  - From lattice transformations $\Lambda_1^t(l)$, symbols from the 2 levels are added to compose the coset partitioning over lattice Z^2 and the final labeling

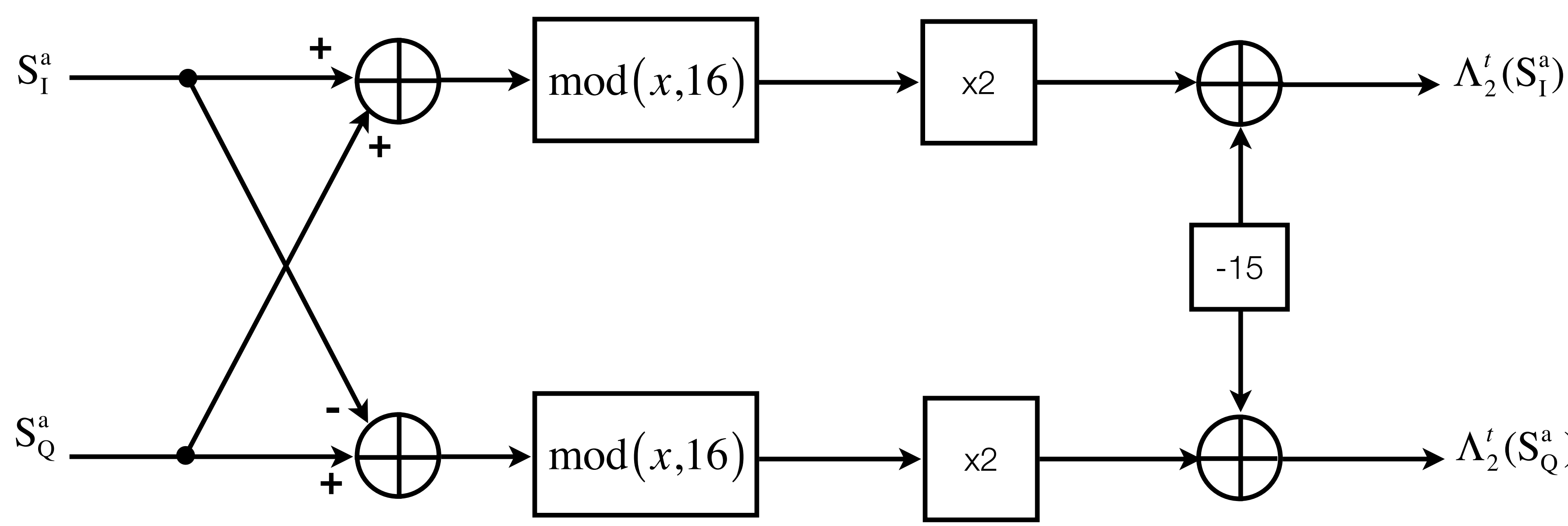


Coded 16-PAM - 2nd lattice transformation

- Lattice transformation Λ_2^t
  - From lattices adder over Z^2, we perform the transformation for obtaining a final zero mean 2D square constellation over RZ^2
  - Transformation is composed by
    - -45º rotation
    - Modulo operation, constraining the constellation points to a square region in the 1st 2D quadrant
    - Centering (zero mean) and scaling

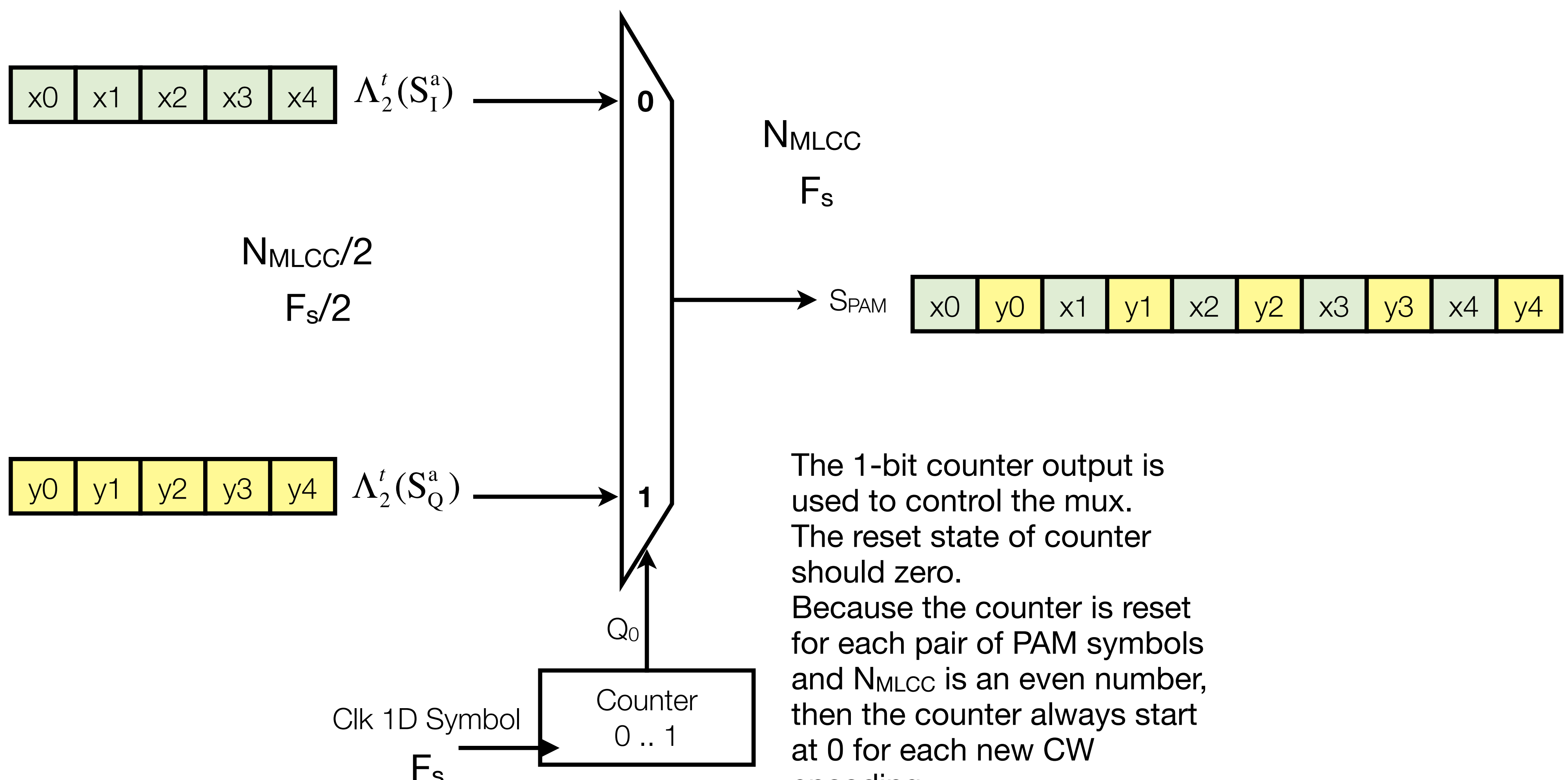
$$\Lambda_2^t(x) = 2\cdot\mathrm{mod}\left(x\cdot(1-j)^{\mathrm{rem}(2\xi,2)}, 2^{\lceil\xi\rceil}\right) + (1+j)\cdot\left(1-2^{\lceil\xi\rceil}\right) \quad \forall x \in \mathbb{C}, \quad j=\sqrt{-1}$$

$$\mathrm{mod}(y,x) \triangleq y - x\left\lfloor \frac{y}{x} \right\rfloor$$



Coded 16-PAM - *RZ*^{2} to PAM multiplexer

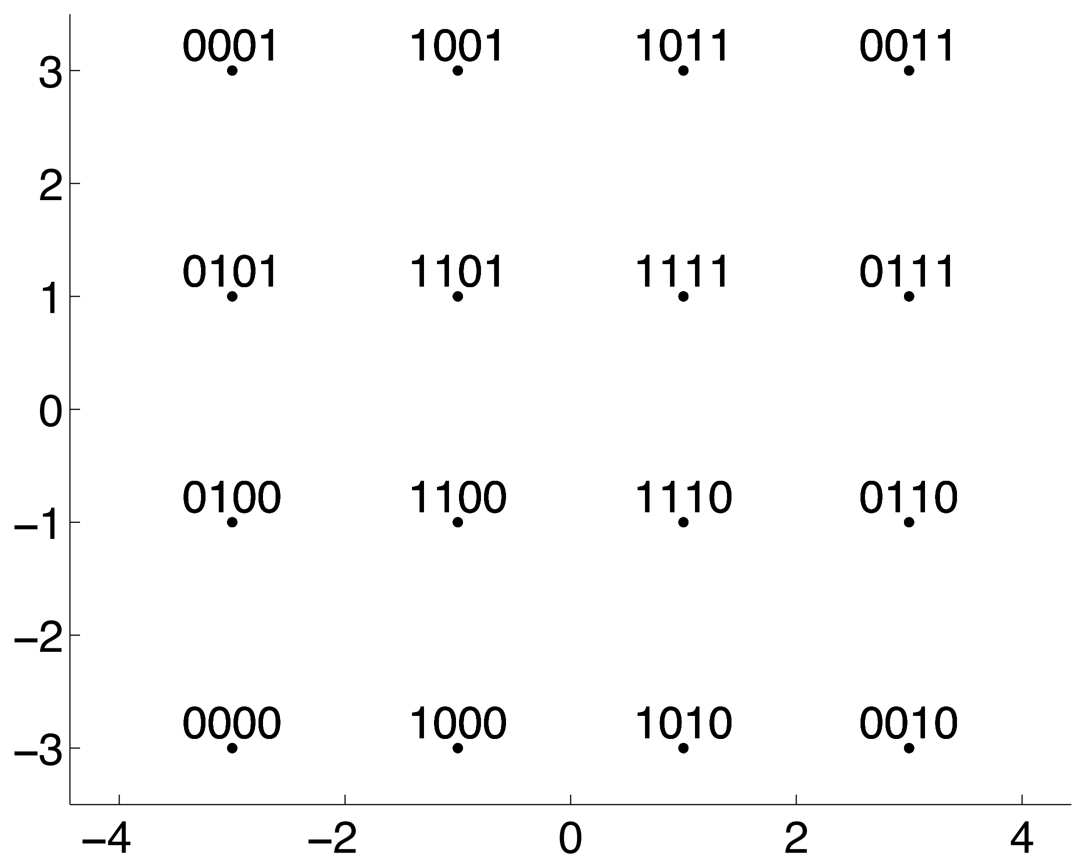
- In-phase and Quadrature components of 2D symbols from Λ_2^t are time-domain multiplexed, composing a sequence of symbols belonging to a 16-PAM constellation
- 16-PAM symbols belongs to the set *{-15, -13, ..., 13, 15}*



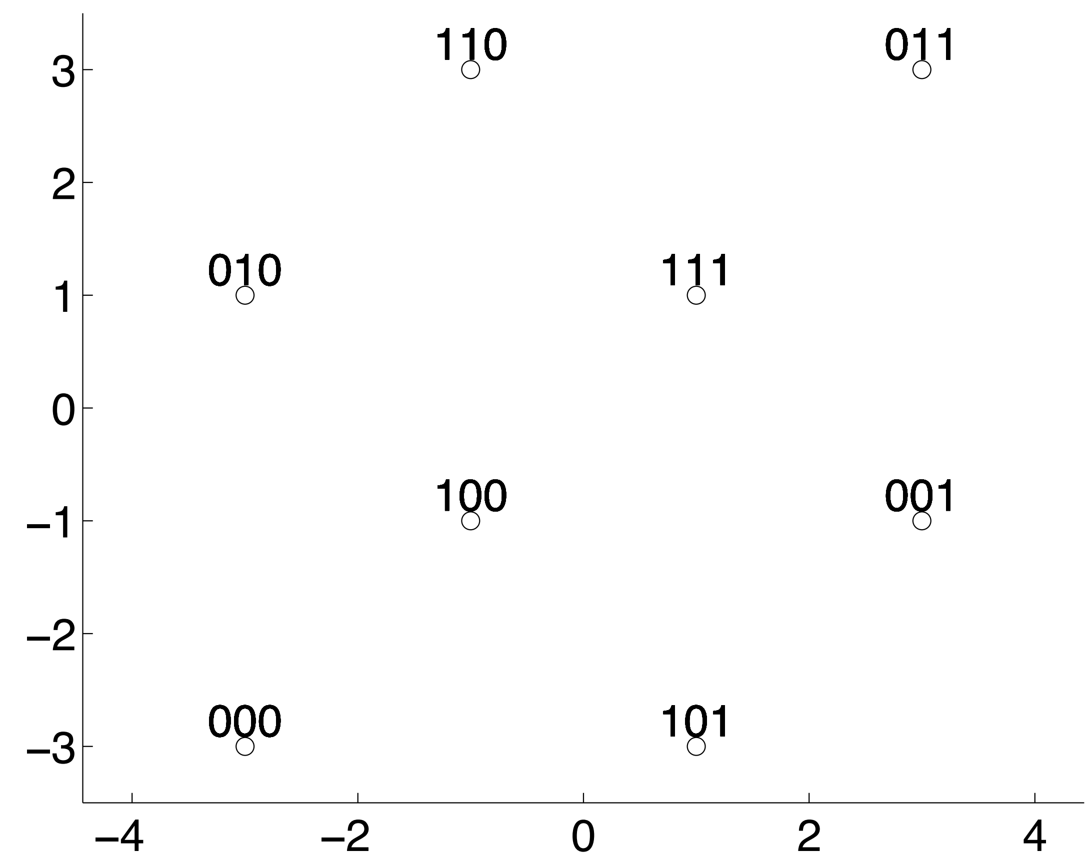
The 1-bit counter output is used to control the mux.
The reset state of counter should zero.
Because the counter is reset for each pair of PAM symbols and N_{MLCC} is an even number, then the counter always start at 0 for each new CW encoding

Coded 16-PAM - labeling and coset partitioning

Level 1, 16-QAM

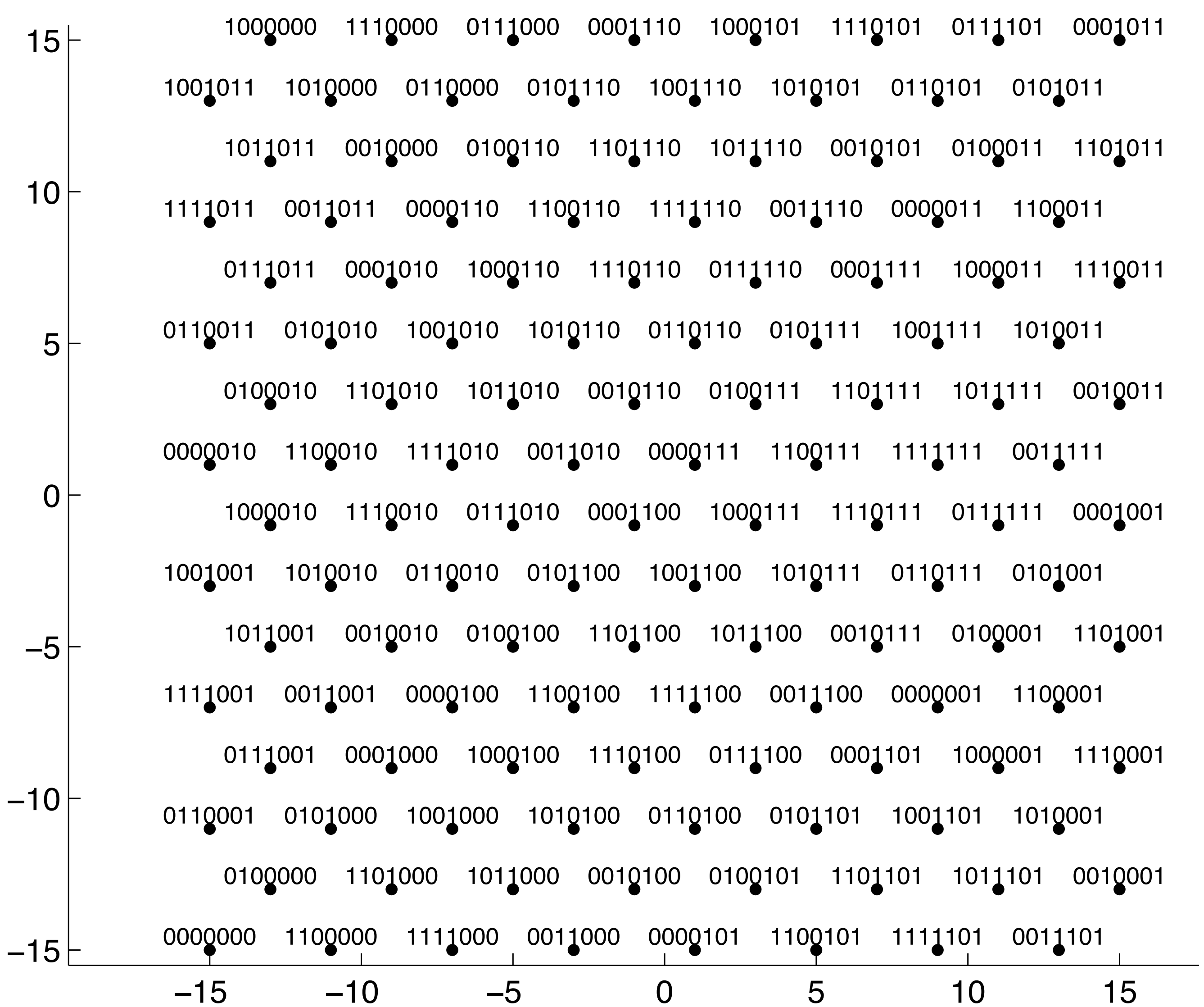


Level 2, 8-QAM



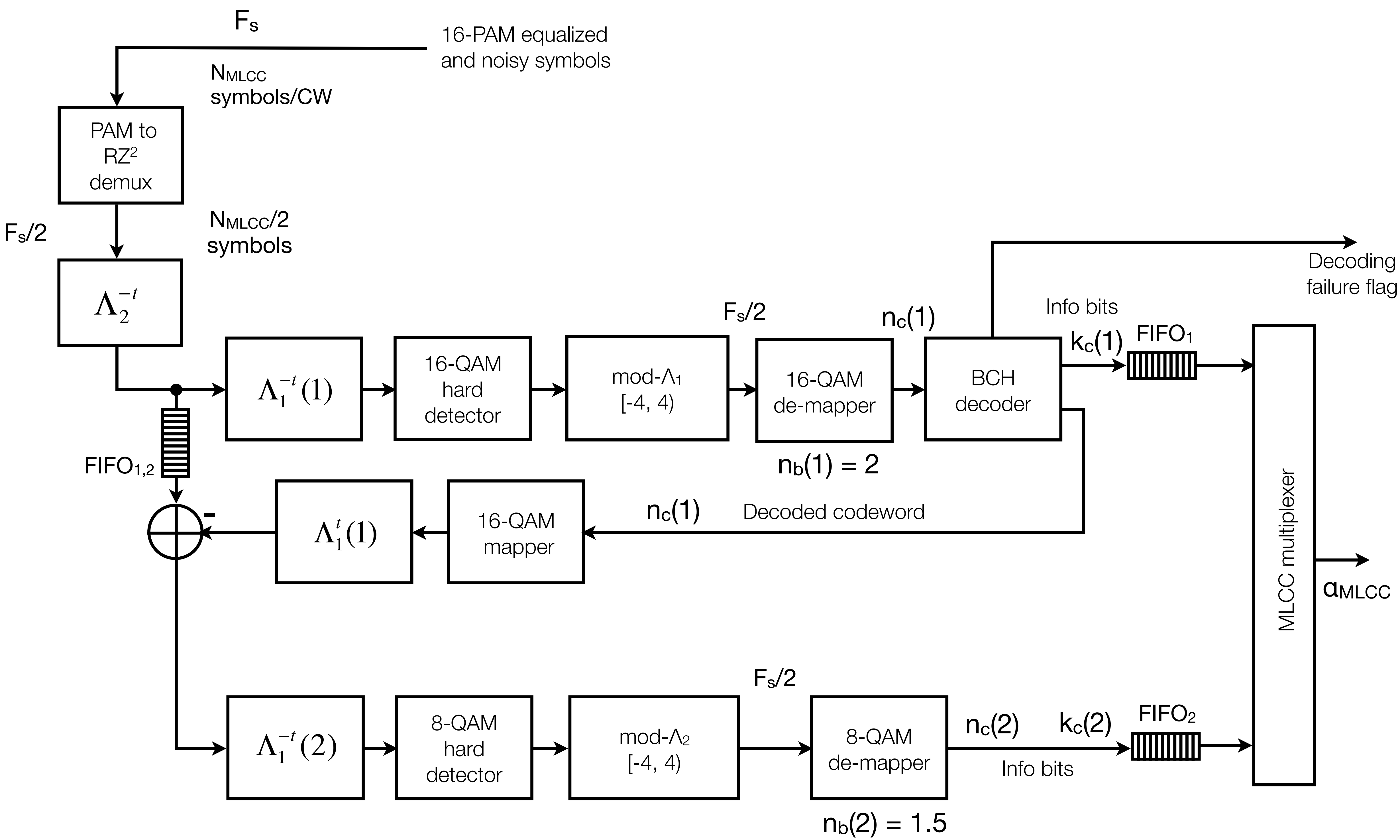
Coded 16-PAM - labeling and coset partitioning

After Λ_2^t, 128-QAM



- Basic numbers of constellation:
  - 128 points in a 2D constellation
  - $\log_2(128)$ = 7 bits / 2D symbol
  - 7 bits =
    - 4 bits of 1st MLCC level
    - 3 bits of 2nd MLCC level
  - Each 2D symbol are transmitted at a rate of Fs/2
  - To transmit over 1D (i.e. intensity modulation of LED), the system does time interleaving of both coordinates of 2D constellation at double rate, that is Fs
  - Each 2D point can be represented by 2 coordinates that can take 16 different values each one: {-15, -13, … 13, 15} ➤ 16-PAM
  - This is 16-PAM, but encoding by 3.5 bits/1D symbol (i.e. 7bits/2D) instead of 4 bits as usual, since the 1D constellation was generated from odd bits 2D constellation.
  - 3.1883 bits of 3.5 are information bits, the rest is parity for error correction

Coded 16-PAM - Multi-Stage Decoder (MSD)



References

- [1] *Rubén Pérez-Aranda, "Shannon's capacity analysis of GEPOF for technical feasibility assessment", GEPOF SG, Interim Meeting, May 2014*
- [2] *Rubén Pérez-Aranda, "High spectrally efficient coded modulation schemes for GEPOF technical feasibility", GEPOF SG, Plenary Meeting, July 2014*
- [3] *G. D. Forney, Jr. et al., "Sphere-Bound-Achieving Coset Codes and Multilevel Coset Codes", IEEE Trans. on Inform. Theory, vol. 46, pp. 820-850, May 2000*

Questions?