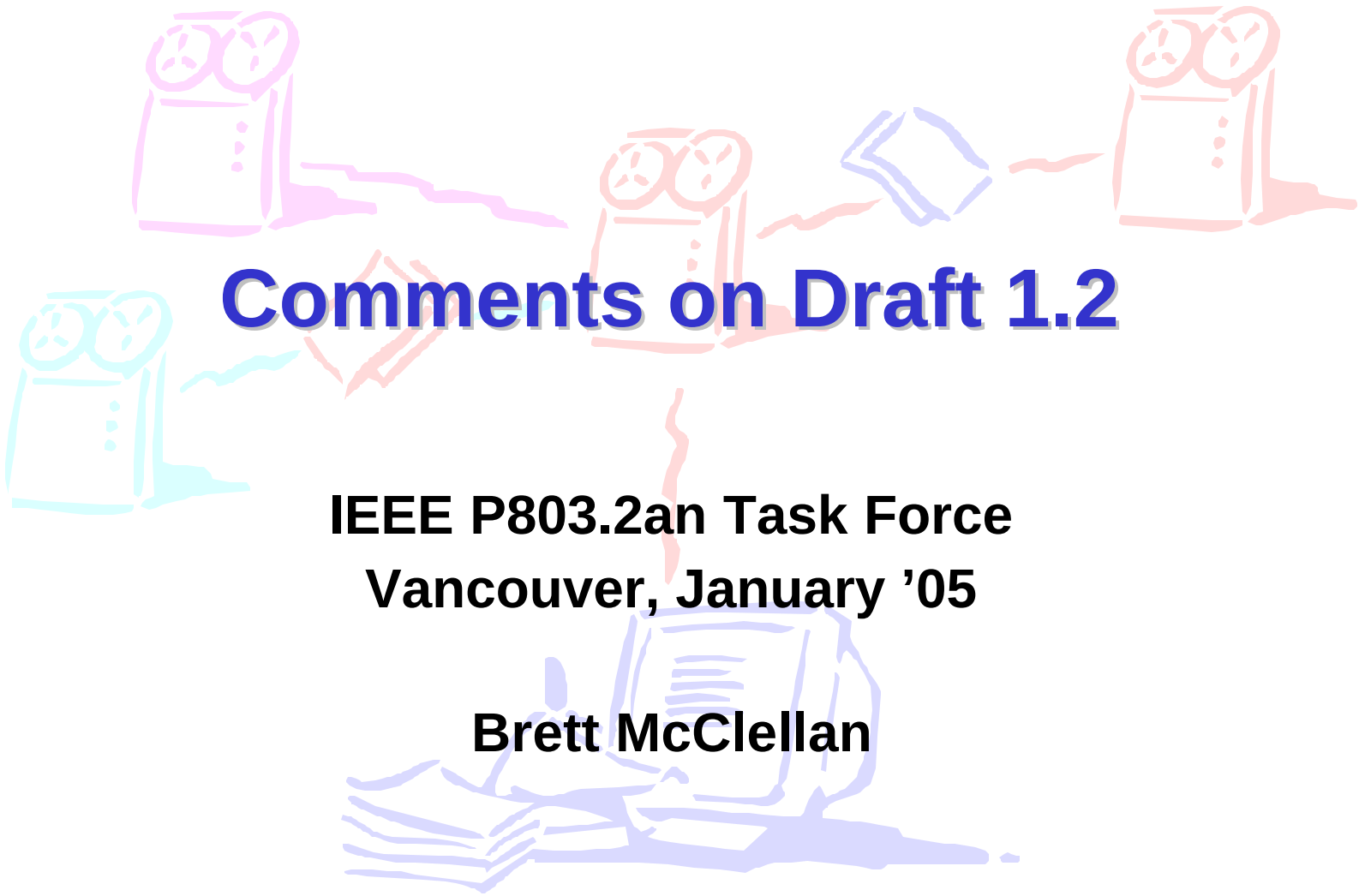


Comments on Draft 1.2

IEEE P803.2an Task Force
Vancouver, January ’05

Brett McClellan

Efficient Encoder Proposal

- I propose that the standard specify a LDPC matrix that provides for efficient encoder implementations.

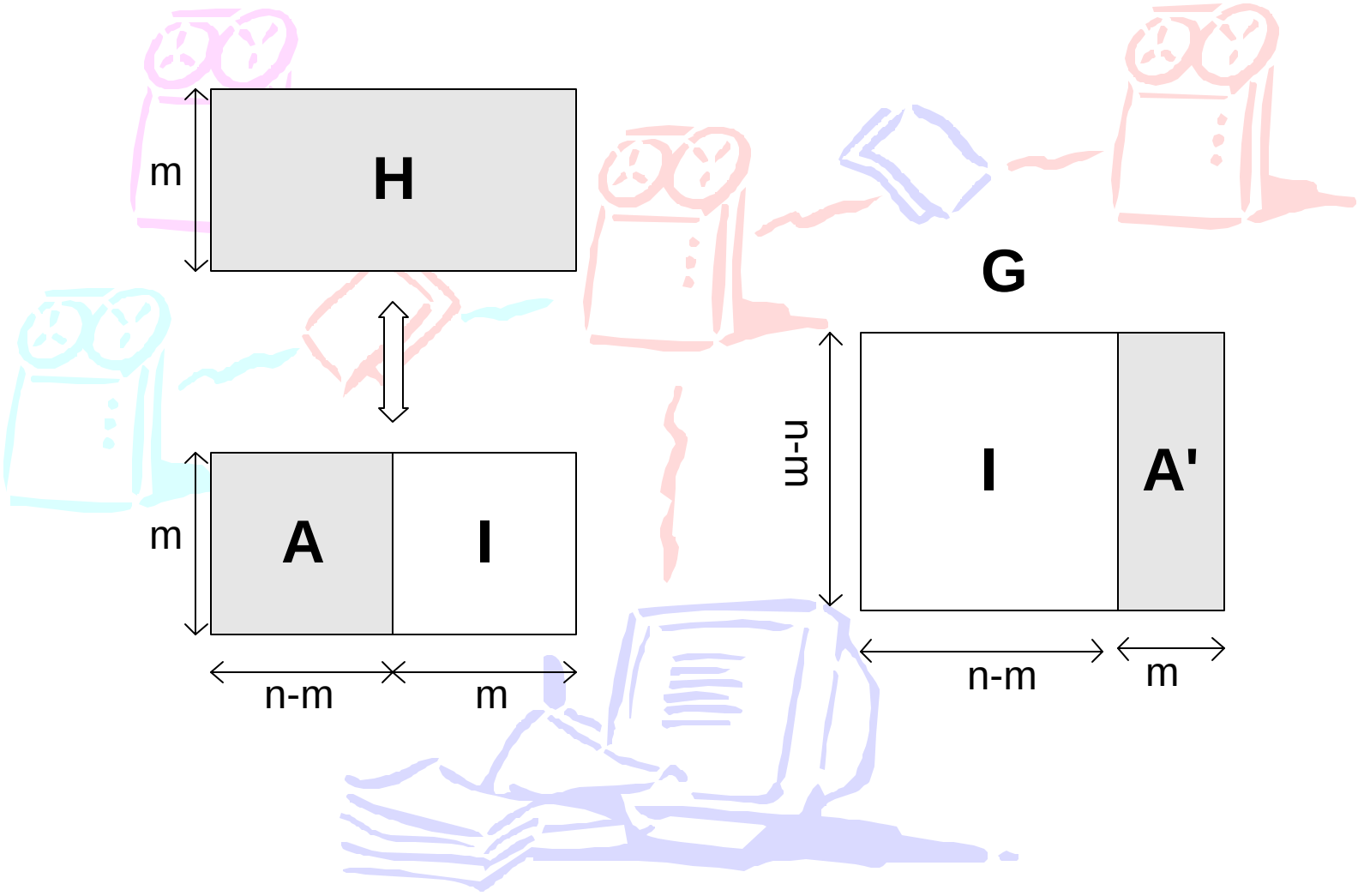
- Outline
  - General Approach
  - Richardson's Method
  - (2048,1723) Example
  - Summary

- Thanks to Michael Leung (Solarflare)

General Approach

- Start with the original parity check matrix: H
  - Manipulate using column swaps and row operations
    - Convert H to [A | I], where I is the identity matrix
    - A is not sparse
- Calculate the systematic generator matrix
  - G = [I | A'], where A' is the parity matrix
- Encode the message symbol: s
  - c = s • G, where c is the code word
    s the systematic bits
- Example for (2048,1723)
  - H is 384 x 1723 and is sparse -> ~0.4%
  - A' is 1723 x 325 and is dense -> ~50%
  - c computation requires ~280k XOR's (>0.5M gates)

General Approach

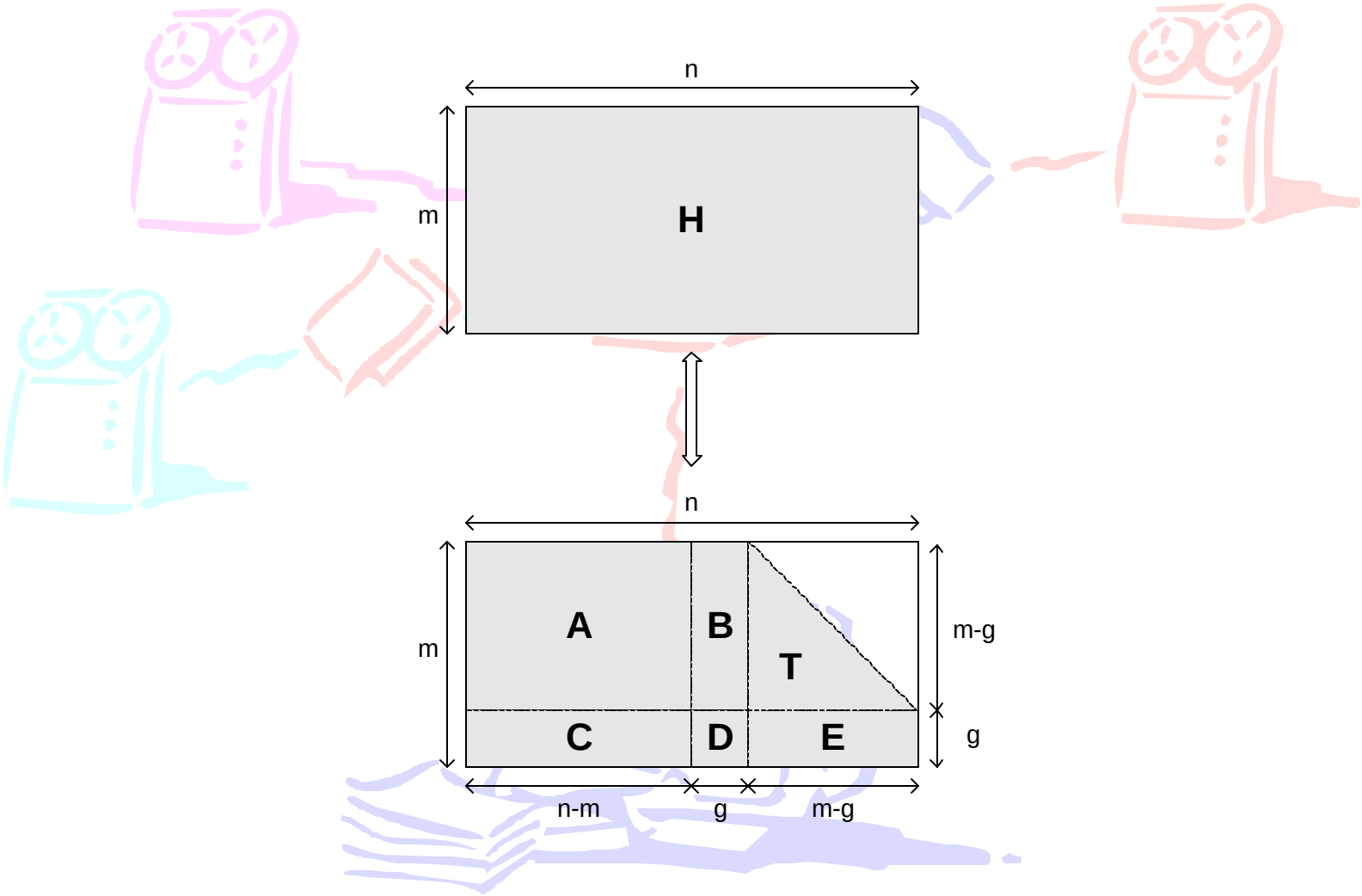


Richardson's Method

- Steps:
  - Bring H into approximate lower triangular form using row and column swaps only.
    - Still a sparse matrix
  - Parse H into six submatrices: A,B,C,D,E & T
    - Where T is lower triangular
  - Calculate the parity check bits by performing a series of efficient computations

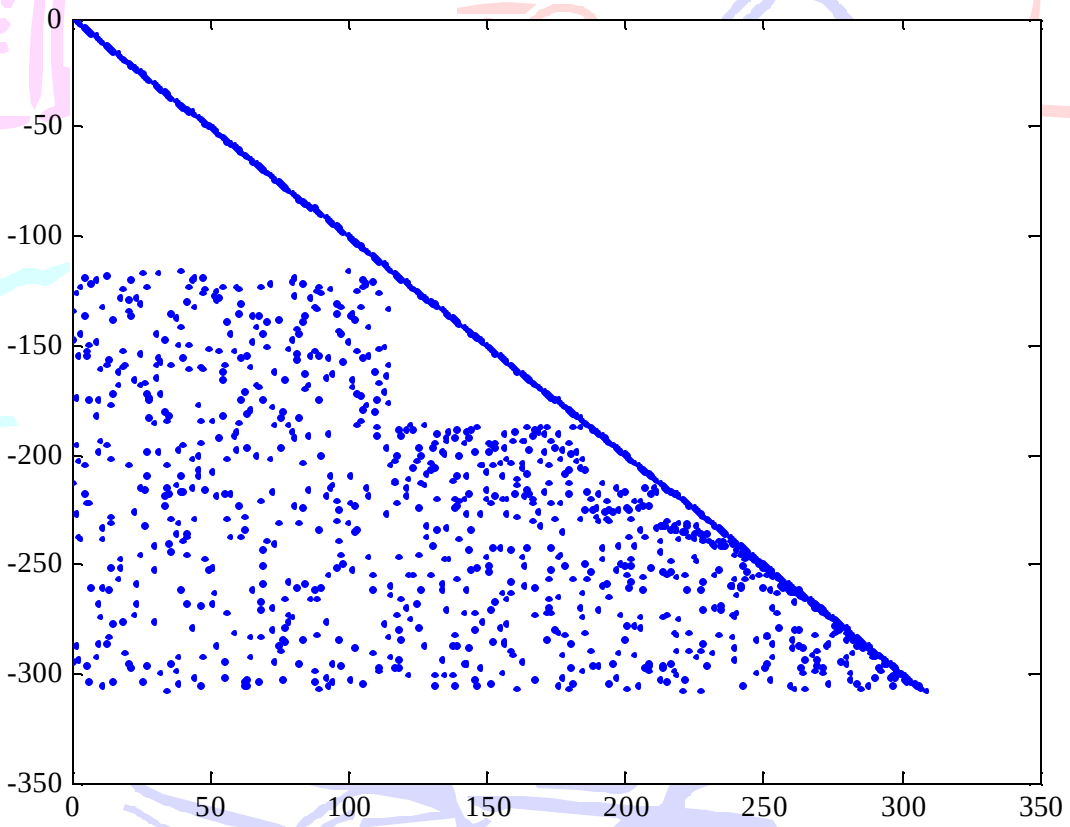
- Reference:
  - Efficient Encoding of Low-Density Parity-Check Codes", T. Richardson and R. Urbanke, IEEE Trans. Info Theory, vol. 47, no. 2, pp. 638-656, Feb. 2001

Richardson's Method



Richardson's Method

- T submatrix – lower triangular



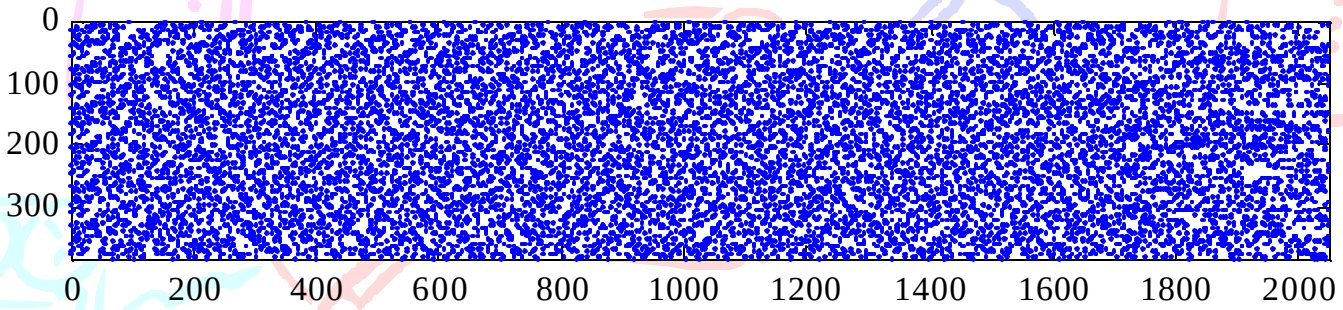
Parity Calculation

- c = (s,p1,p2)

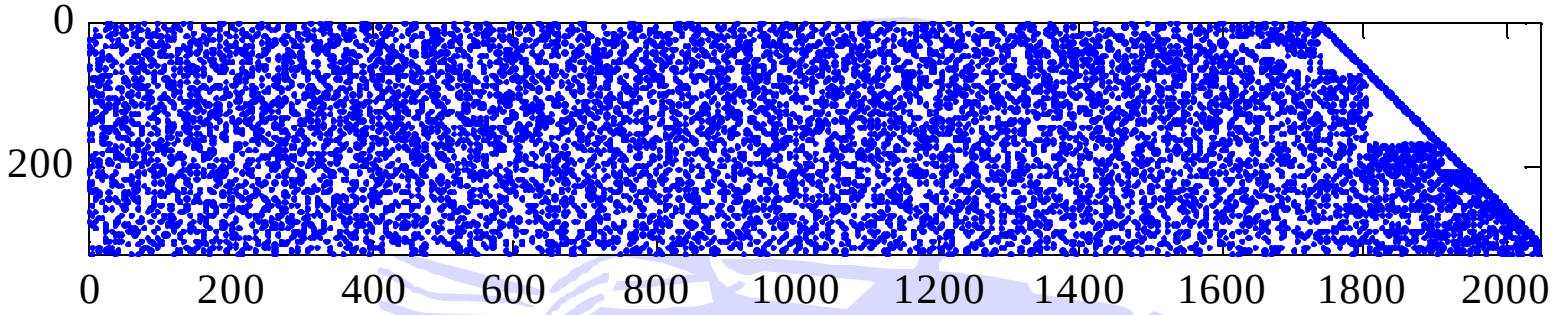
Vector	Formula	Method	Complexity (Xor)
i_1	$i_1 = A\ s'$	Direct Computation	8544
i_2	$i_2 = T^{-1}\ i_2$	Back substitution => $i_2 = T\ i_3$; as T is sparse and triangular	987
i_3	$i_3 = E\ i_3$	Direct Computation	120
i_4	$i_4 = C\ s'$	Direct Computation	359
i_5	$i_5 = i_3 + i_4$	Direct Computation	15
p_1	$p_1 = \Psi^{-1}\ i_5$	Direct Computation, where $\Psi = E\ T^{-1}\ B + D$	111
i_6	$i_6 = B\ p_1'$	Direct Computation	79
i_7	$i_7 = i_1 + i_6$	Direct Computation	310
p_2	$p_2 = T^{-1}\ i_7$	Back substitution => $i_7 = T\ p_2$; as T is sparse and triangular	987

(2048,1723) Example

- **H matrix w/ column swaps**



- **Approximate lower triangular**



Summary

- Richardson's method reduces encoding complexity by more than an order of magnitude.
  - ~280k -> ~12k
- No effect on decoder performance
- Standard should specify H in sparse format
  - 384x32 = 12888 entries
- Standard should also specify the row swap vector to generate A,B,C,D,E & T
  - 384 entries
- The example matrix has been submitted to the editor

PCS Scrambler

- The Clause 49 self-synchronized scrambler is a proven design for 10G Ethernet.
  - $1+X^{39}+X^{58}$
  - Analyzed for CRC interaction and jamming susceptibility.
    - http://www.ieee802.org/3/10G_study/public/jan00/walker_1_0100.pdf
  - At most, the error propagation in 10GBASE-T will be a single 65B block following an LDPC frame error.
    - Same propagation as 10GBASE-R

PCS Scrambler cont.

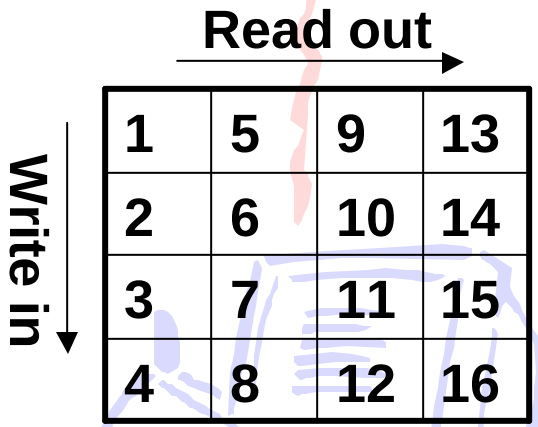
- For 10GBASE-T
  - Slave should transmit using $1+X^{19}+X^{58}$ polynomial
    - time reversed sequence
  - CRC added after scrambler / checked before descrambler
    - No error propagation in CRC check
  - Proposed “sync” bit can be traded for CRC8 vs CRC7
    - $(1+X+X^5+X^6+X^8)$
  - Proposed “aux” bit can be traded for CRC9
    - $(1+X^2+X^3+X^5+X^{10})$

PCS Scrambler cont.

- If error propagation is a concern:
  - Change to sidestream scrambler using the same polynomial.
    - Master: $1+X^{39}+X^{58}$
    - Slave: $1+X^{19}+X^{58}$
  - Synchronize the descrambler during training/data transition
    - Use Aux bit for simple handshaking
    - Master and Slave transmit 0's instead of 65B data w/ Aux=0
    - Master synchronizes descrambler to received sequence
      - Transmits scrambled 0's w/ Aux=1
    - Slave synchronizes descrambler to received sequence
      - Transmits scrambled 65B data w/ Aux=1
    - Master responds
      - Transmits scrambled 65B data w/ Aux=0
    - Slave responds
      - Transmits scrambled 65B data w/ Aux=0

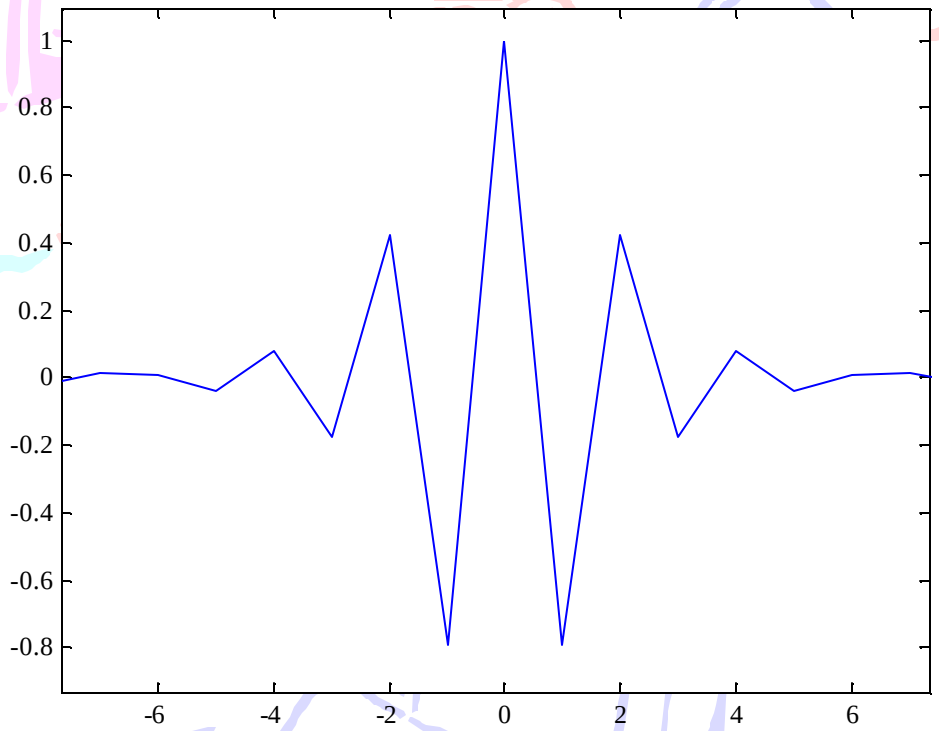
Symbol Interleaver

- **Noise correlation at the equalizer output degrades the uncoded bit error performance.**
  - **DSQ symbols are mapped using time adjacent samples**
- **A simple block interleaver can compensate**
  - **4x4 block interleaver on each pair**



Example

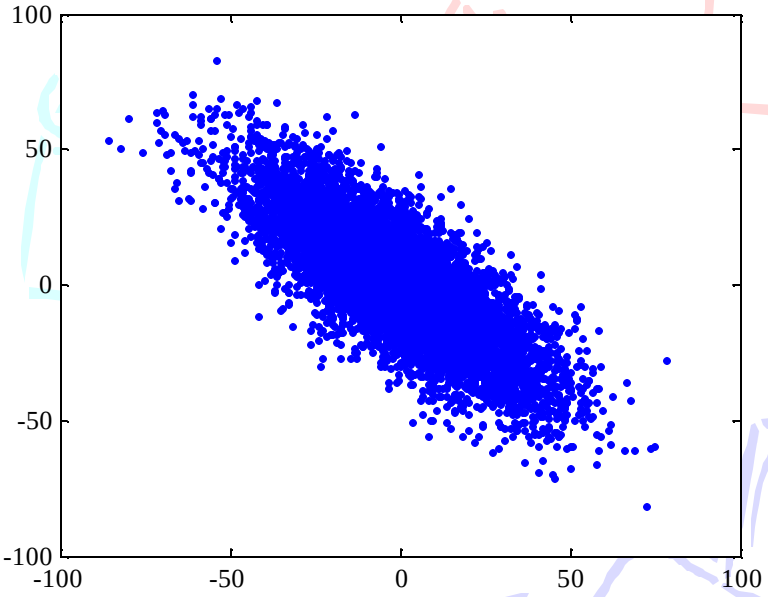
- **Extreme noise correlation at EQ out:**



Example

- **In extreme case, interleaver reduces noise by 3dB**

Interleaver In



Interleaver Out