Flash Memory: Signal Processing Perspective

Jaekyun Moon

EE Dept
KAIST

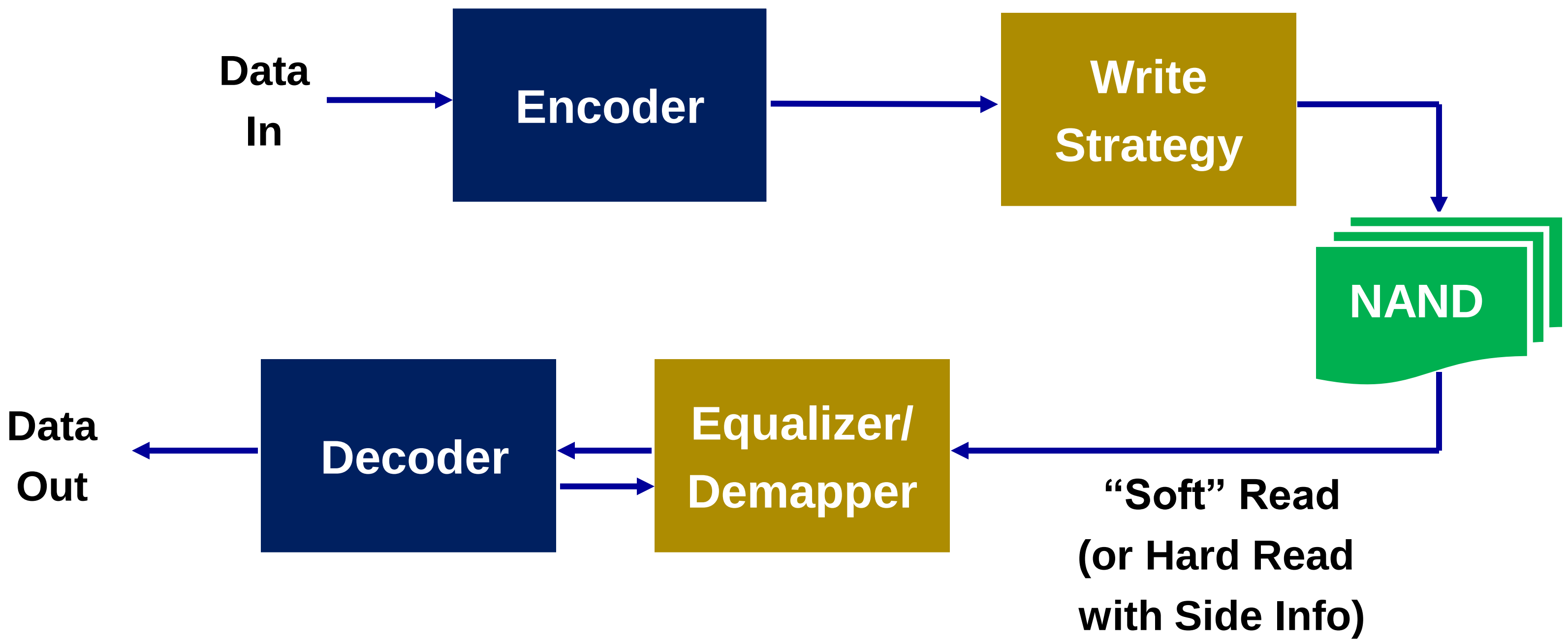
April 20, 2010

NVRAMOS Forum

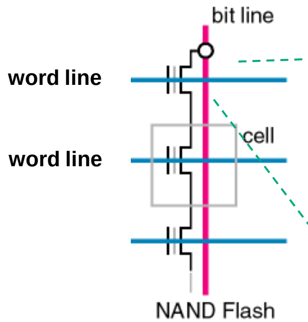
Outline

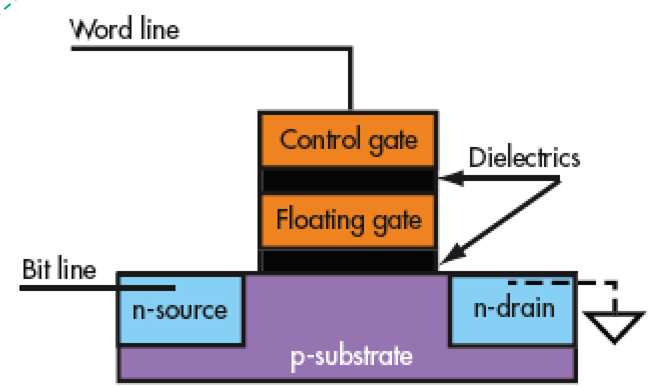
- Review of the write, read, erase and disturb at the device level
- Why LDPC code?
  - performance potential in comparison with BCH
- Set-partitioning aided by soft information and multi-level coding
- Soft demapping (cell output to bit-level soft decision conversion)
- Soft information evolution in LDPC decoding
- Disturb modeling and turbo equalization

SSP Signal Flow



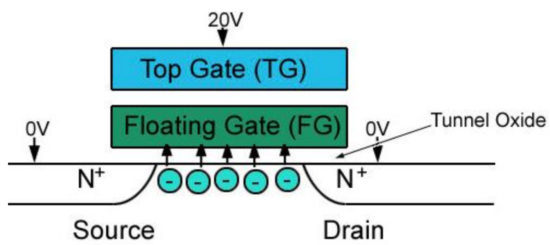
Basic NAND Cell Structure



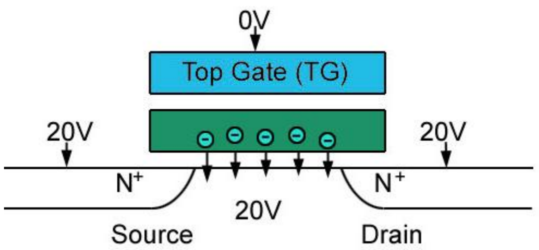


When a positive potential is applied to the control gate in a typical flash memory cell, negatively charged electrons are pushed through the thin oxide layer between the substrate and the floating gate and are trapped on the floating gate. This potential alters the threshold value.

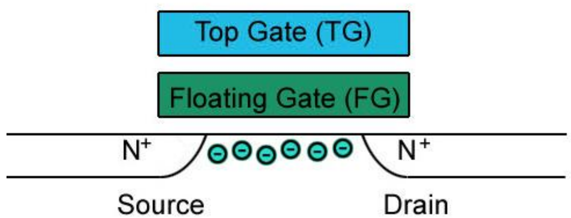
Write, Read and Wear (Endurance)



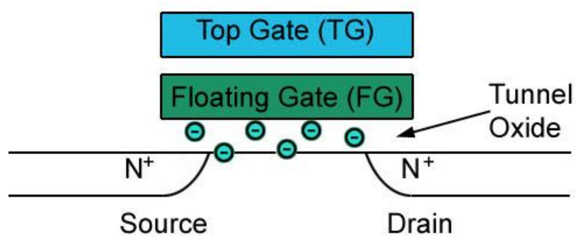
Programming



Erase



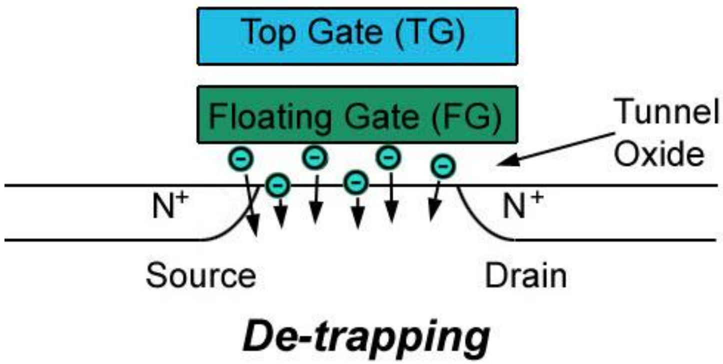
After erasure for a new cell:
no stuck electrons in the tunnel oxide layer

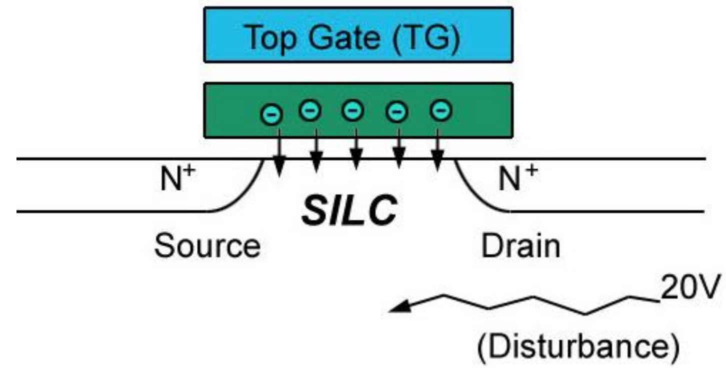


After erasure for an end-of-life cell: with increasing programming cycles, more and more electrons getting trapped in the tunnel (erasure eventually becomes impossible)

Source: Michael Schuette , *The Brave New World of SSDs*, www.lostcircuits.com, 2009

Detrapping & SILC



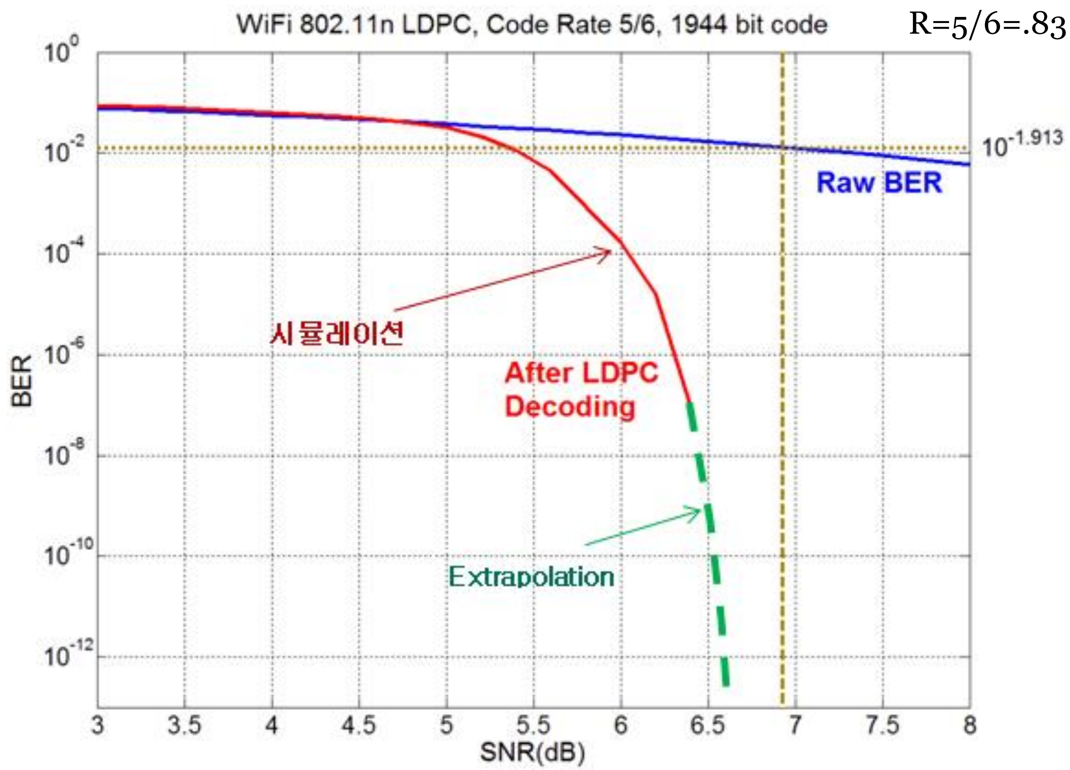


Temperature-cycling and slow cycling can cause de-trapping of stuck electrons in the tunnel oxide and layer boundaries.

Disturbance and other stress factors can cause stress-induced leakage current (SILC), the escape of charge from the floating gate into the substrate. SILC can be a result of programming or reading nearby cells.

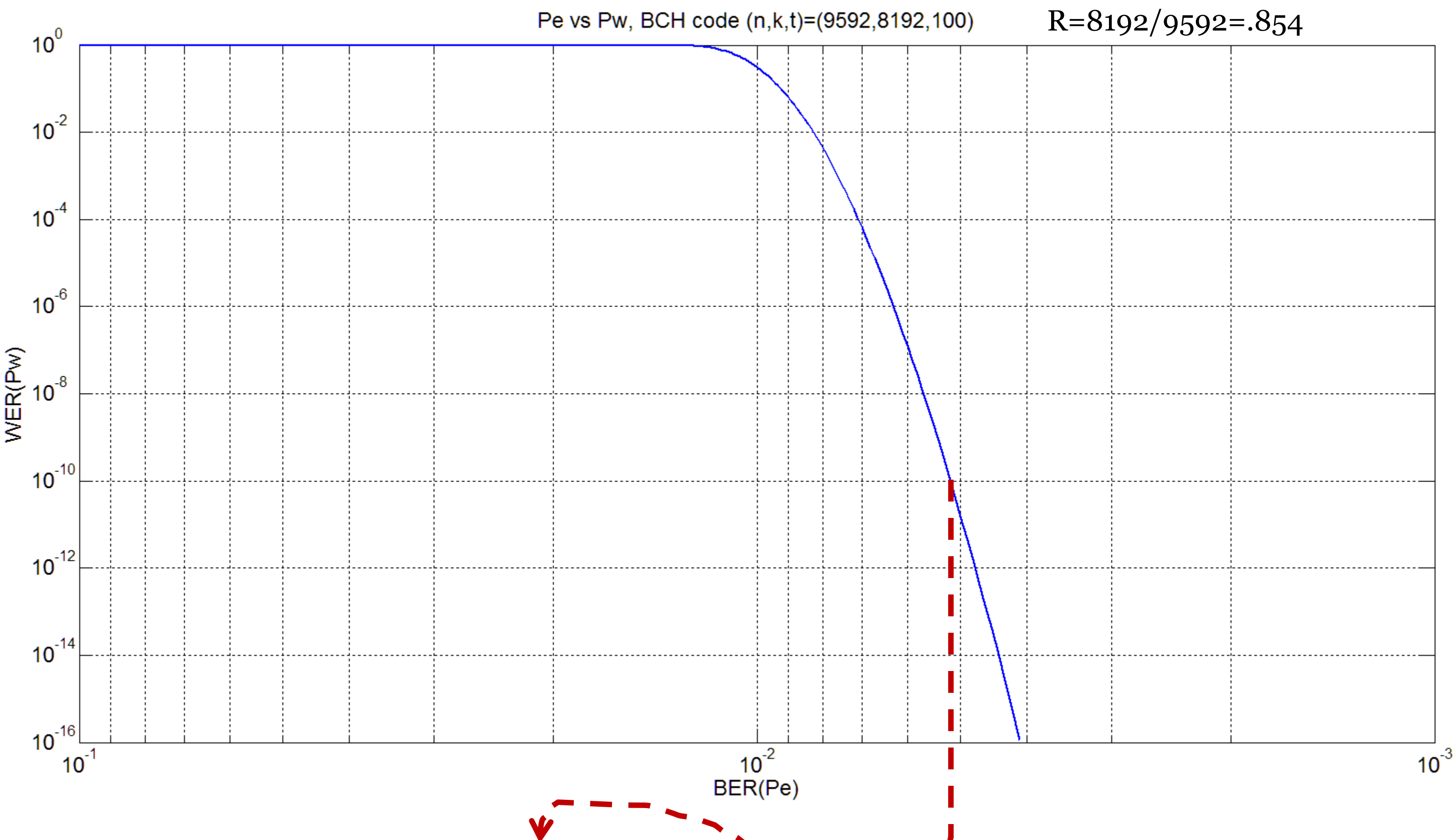
Under SILC, a read disturb can affect cells in an addressed word-line (resulting in electron injection in erased cells during read operation).

LDPC Code Performance



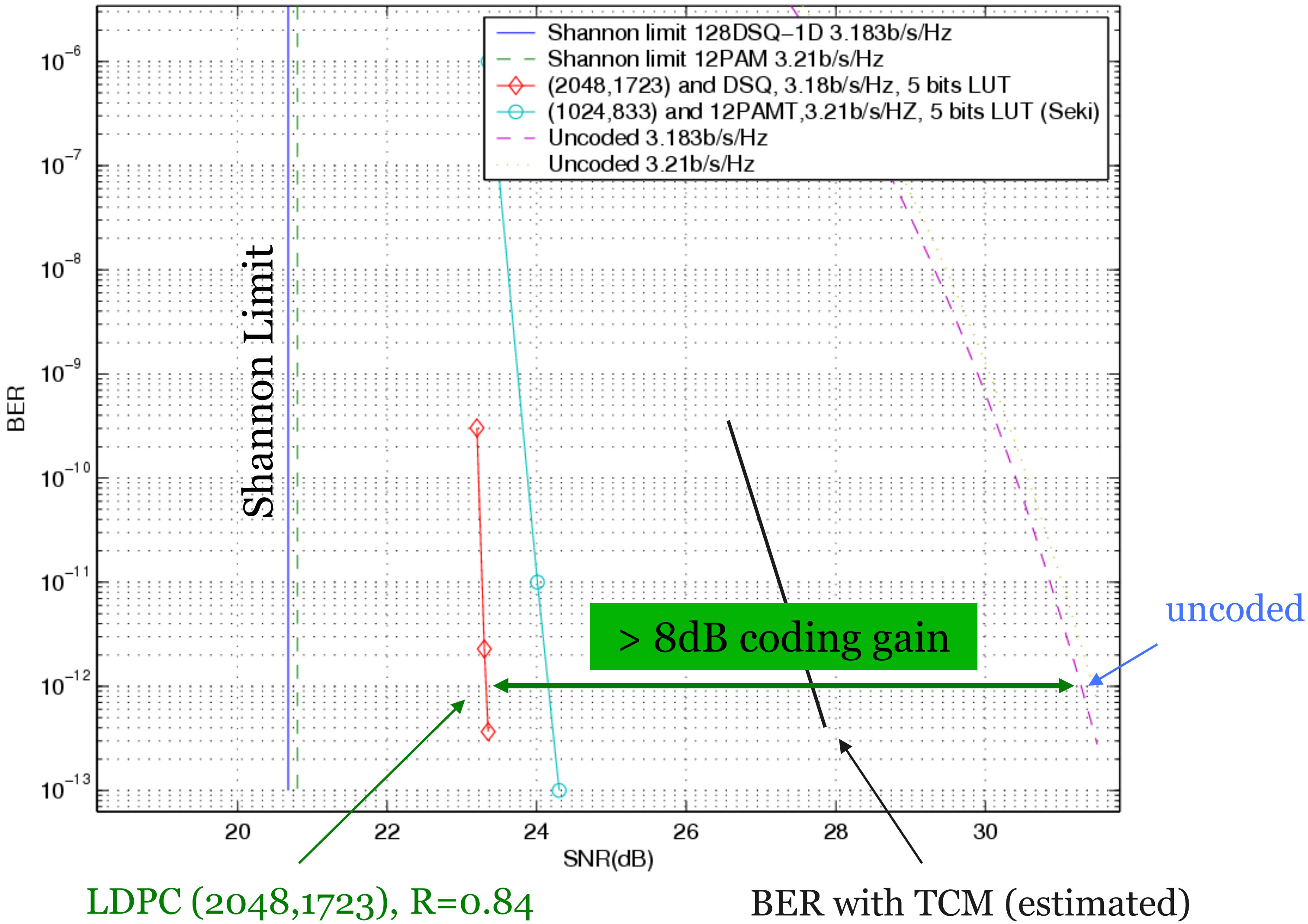
LDPC can handle 10^(-1.913)*9830=120 error bits on average per 9830 bits.

BCH Code Performance: Raw BER vs Corrected WER



Need BER of ~5x10^(-3) or lower
to bring down WER to less than 10^(-10).
Can handle less than 48 error bits on average per 9592 bits.

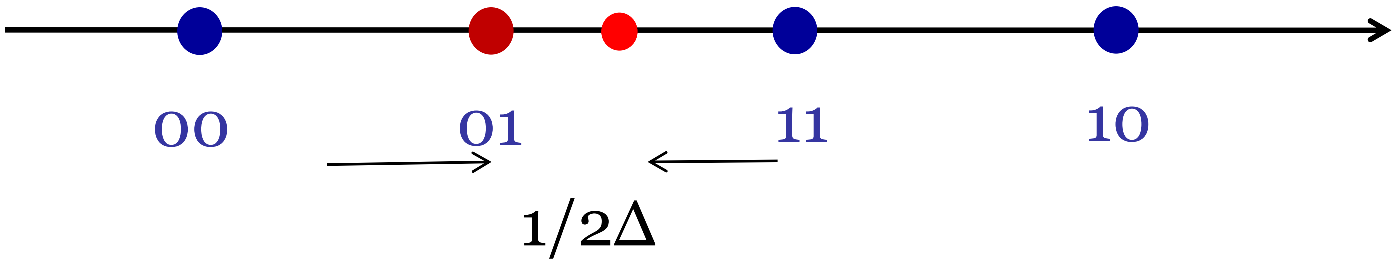
Real LDPC Error performance (10G Ethernet) – no visible error floor (give us hope!)



Source: G. Ungerboeck, 2010

Set-Partitioning Aided by Side Information

MLC: Raw BER for Gray-Mapping and Hard Decision



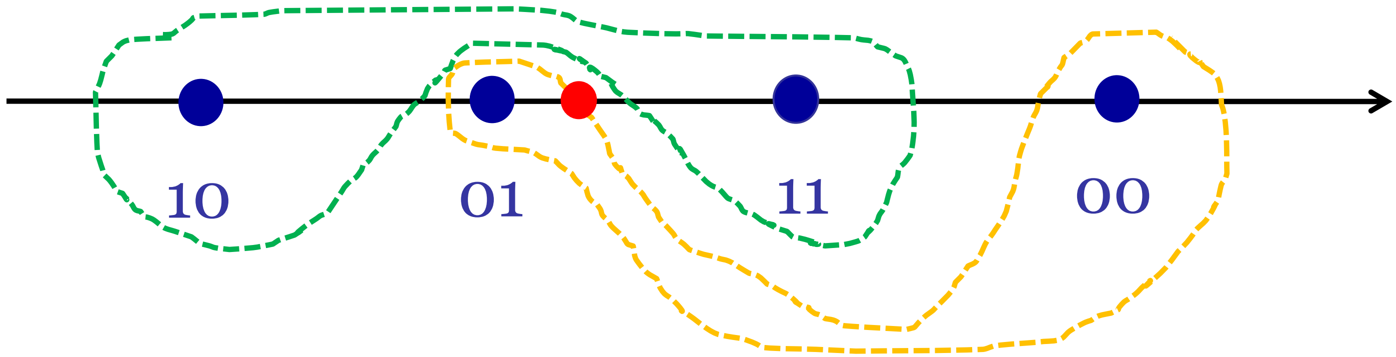
$$P_{raw} = \frac{3}{4} P(n > \Delta / 2)$$

(n-s, k-s, t)=(8751,8192,40) BCH Code

$$P_{word\ BCH} = P(error > t) = \sum_{i=41}^{n=8751} \binom{n}{i} P_{raw}^{i} (1 - P_{raw})^{n-i}$$

Multi-Level Coding: Set-Partitioning

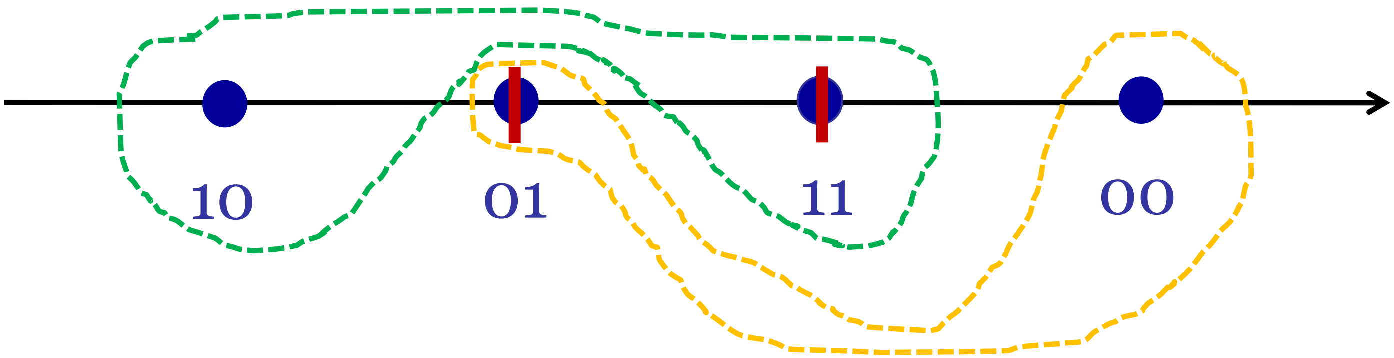
set-partitioning:



Example:
Assume 11 is the correct signal
but noise forces a 01 read.
After ECC correction on MSB, we have the correct group label 1.
Between 10 and 11 in group 1, hopefully 11 will finally be decided.
(utilize some sort of side information such as 01+ or 01- read, if available).

Multilevel-Code Error Rate

set-partitioning:



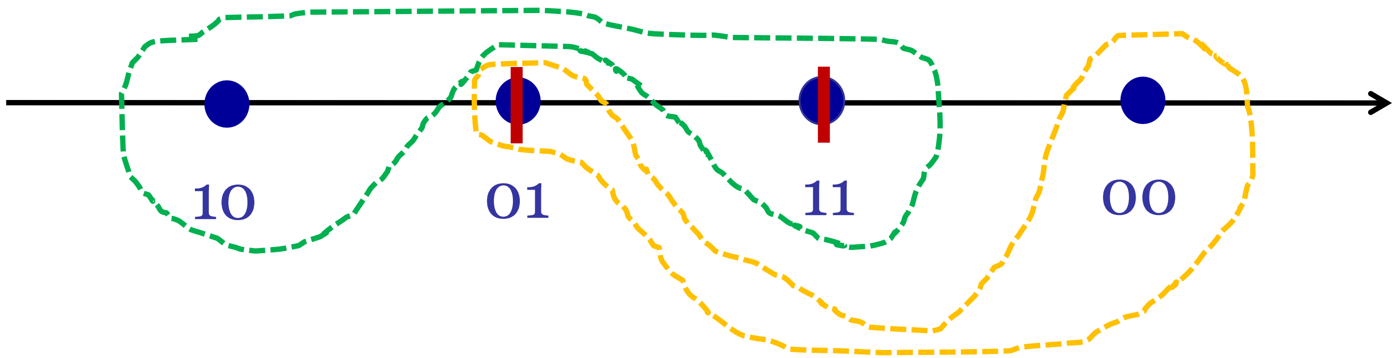
$$P_{word\ MLBCH} = P(\text{LSB in error}|\ \text{MSB corrected})P(\text{MSB corrected}) + \underbrace{P(\text{LSB in error}|\ \text{MSB not corrected})}_{1} P(\text{MSB not corrected})$$

$$\approx \underbrace{P(\text{LSB in error}|\ \text{MSB corrected})}_{P(n>\Delta)} + P(\text{MSB not corrected})$$

$$P(\text{MSB not corrected}) = \text{BCH failure probability with } P_{raw} = 1.5P(n > \Delta / 2)$$

LSB detection signal margin improved by 6 dB

Hard Information Plus Side Information



Channel Output:

10

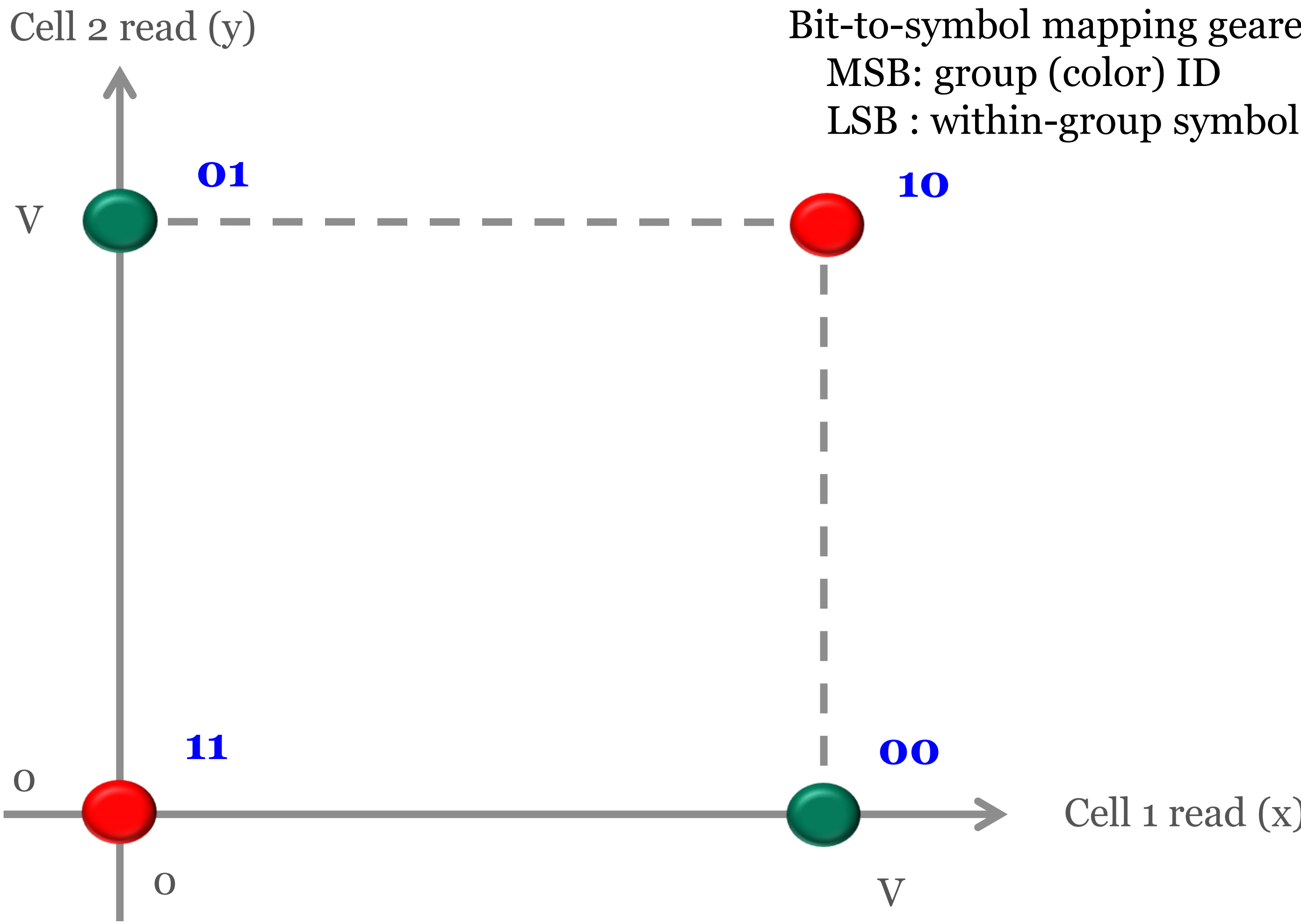
01 (01+ or 01-)

11 (11+ or 11-)

00

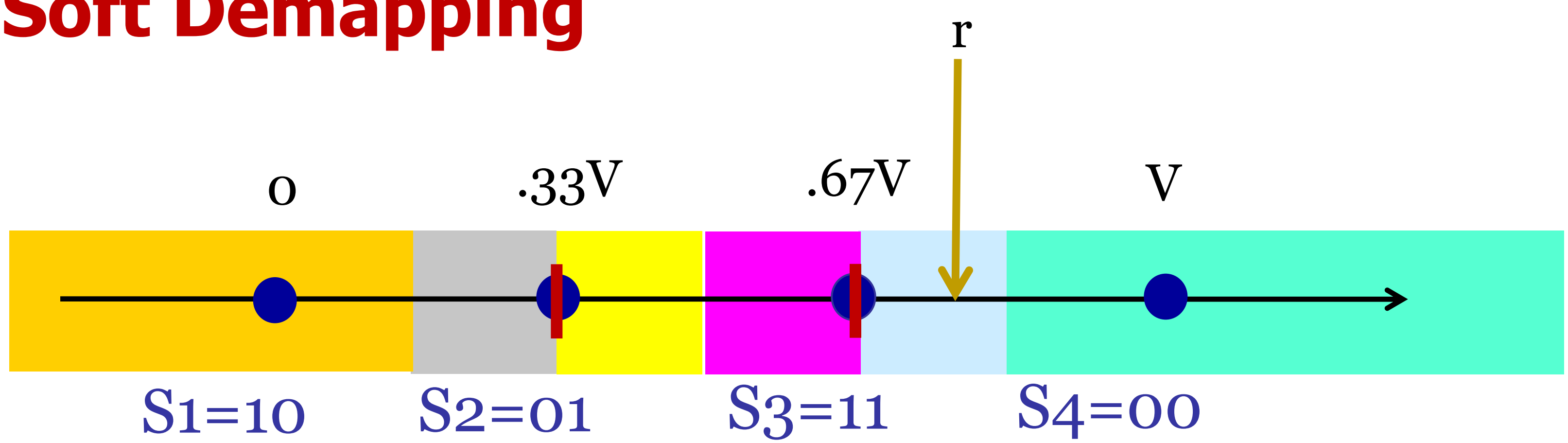
Form a 4-Point Constellation Based on 2 Consecutive Cells …
(Even in MLC or TLC, only 1 bit in each "page-cell")

Bit-to-symbol mapping geared to set-partitioning:
MSB: group (color) ID
LSB : within-group symbol ID



Soft-Demapping

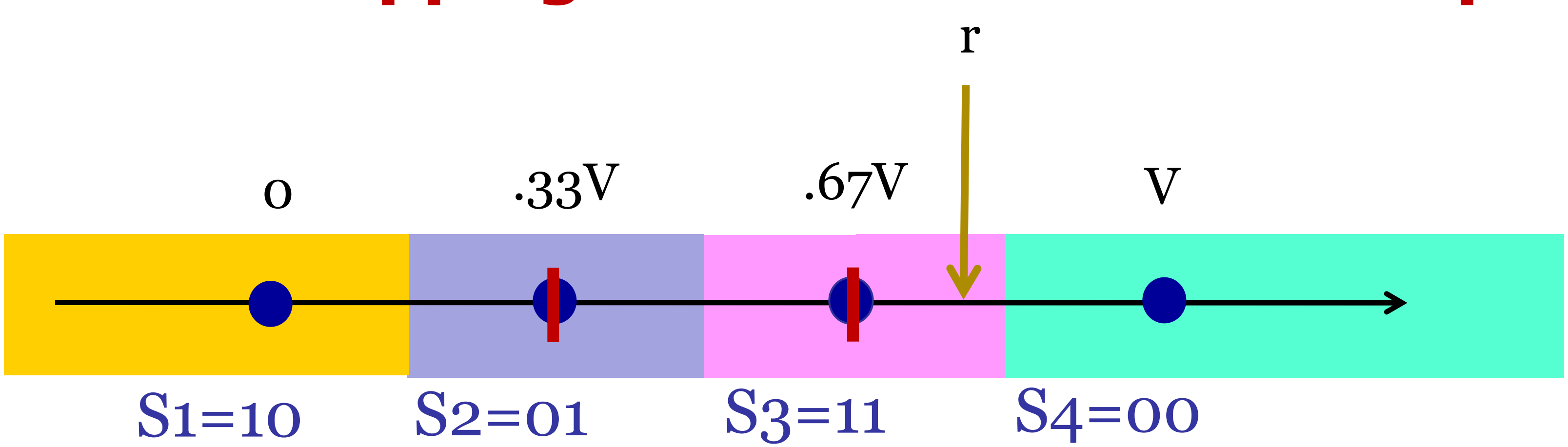
Soft Demapping



Ex: given r ("gray blue"), estimate P(s1), P(s2), P(s3) and P(s4).
P(MSB=1)=P(s1)+P(s3),
P(LSB=1)=P(s2)+P(s3).

Soft decision:
LLR(MSB)=log{P(MSB=1)/P(MSB=0)}
LLR(LSB)=log{P(LSB=1)/P(LSB=0)}

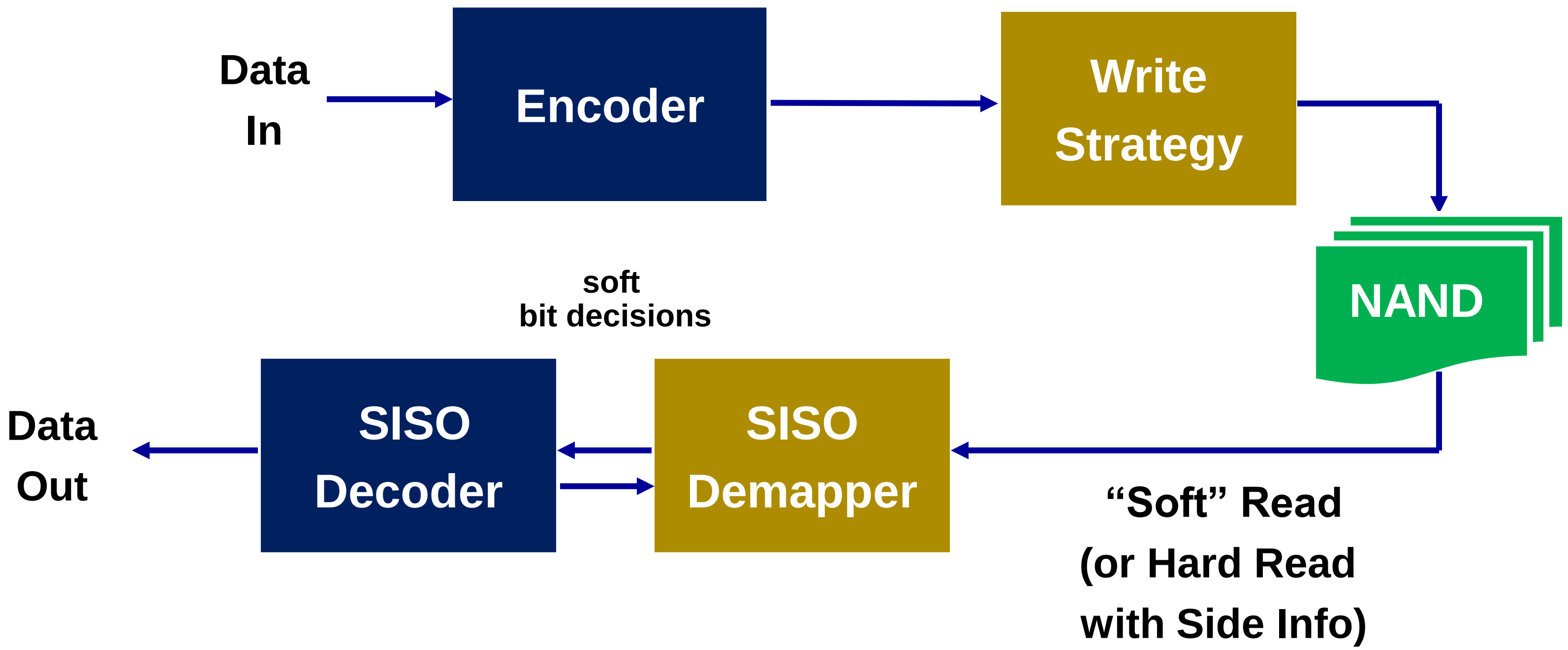
Soft Demapping in the Case of Hard Output



Ex: given r (“pink”), estimate P(s1), P(s2), P(s3) and P(s4).
P(MSB=1)=P(s1)+P(s3),
P(LSB=1)=P(s2)+P(s3).

Soft decision:
LLR(MSB)=log{P(MSB=1)/P(MSB=0)}
LLR(LSB)=log{P(LSB=1)/P(LSB=0)}

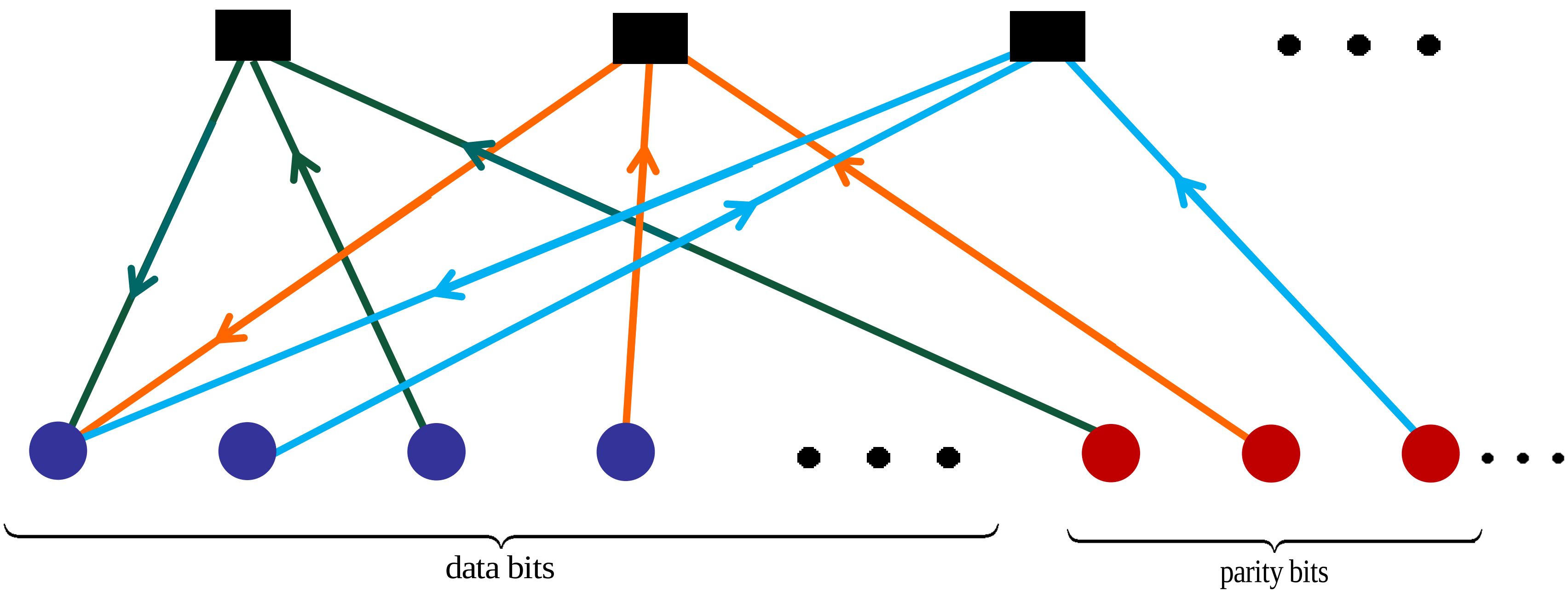
Iterative Soft Information Processing



Ex: given r "quantized", estimate P(s1), P(s2), P(s3) and P(s4).
P(MSB=1)=P(s1)+P(s3),
P(LSB=1)=P(s2)+P(s3).

Estimates of P(s1),…,P(s4) can improve with a priori information on P(MSB=1) and P(LSB=1)

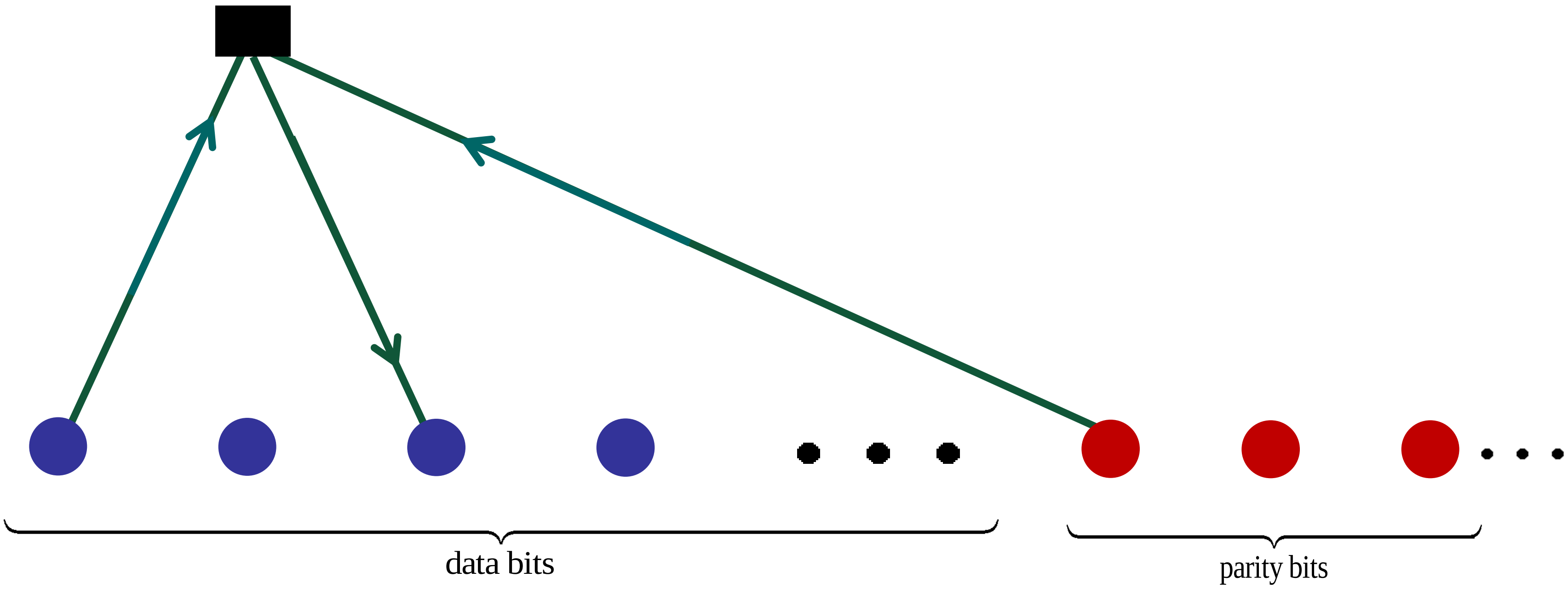
How Does Soft Information Improve During LDPC Decoding



Strong 1
Weak 1
Medim 1

Strong 1
Weak 1
Weak 1

Strong 0
Weak 0
Strong 0

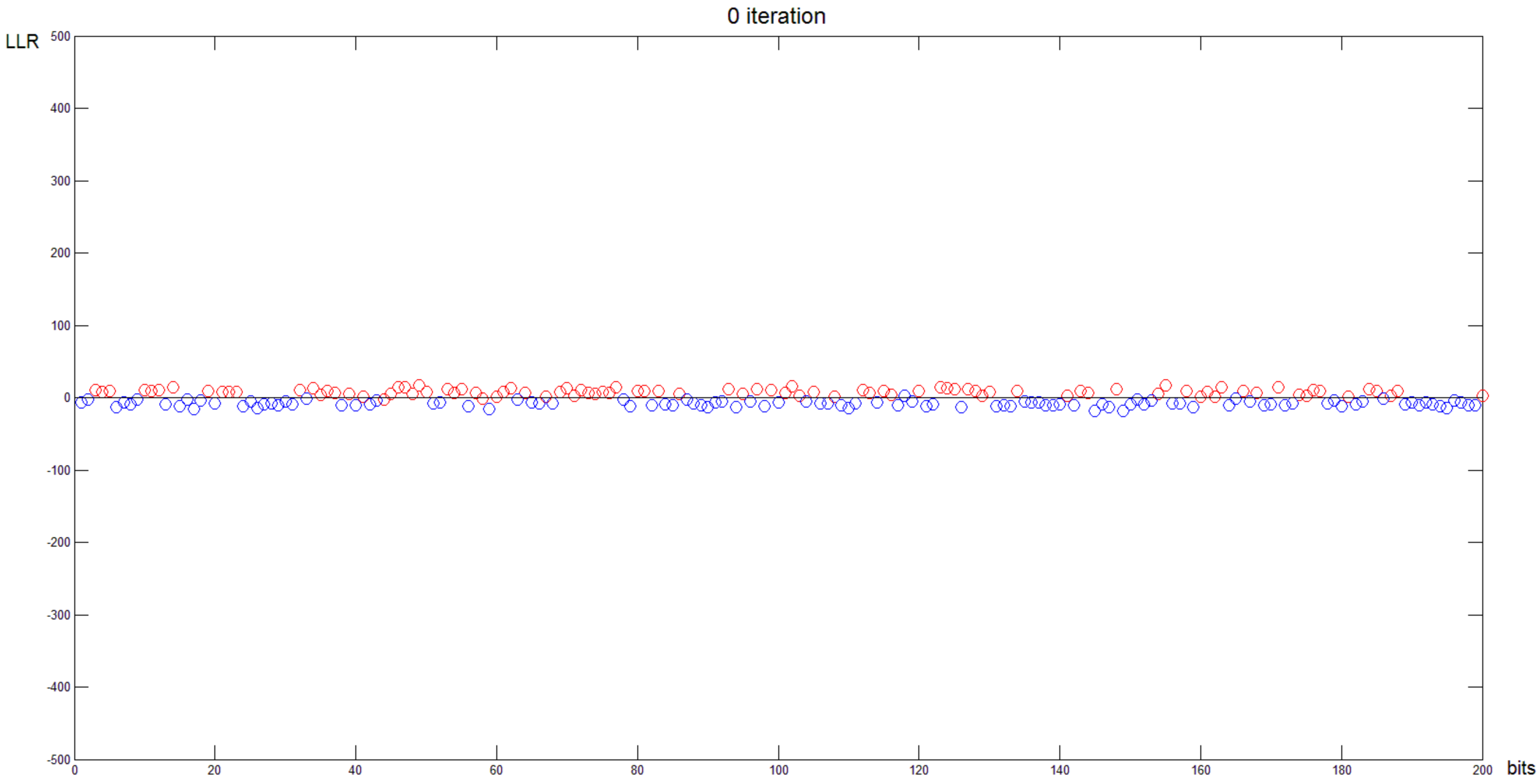

data bits
parity bits

Simulation Results

LDPC Code Parameters

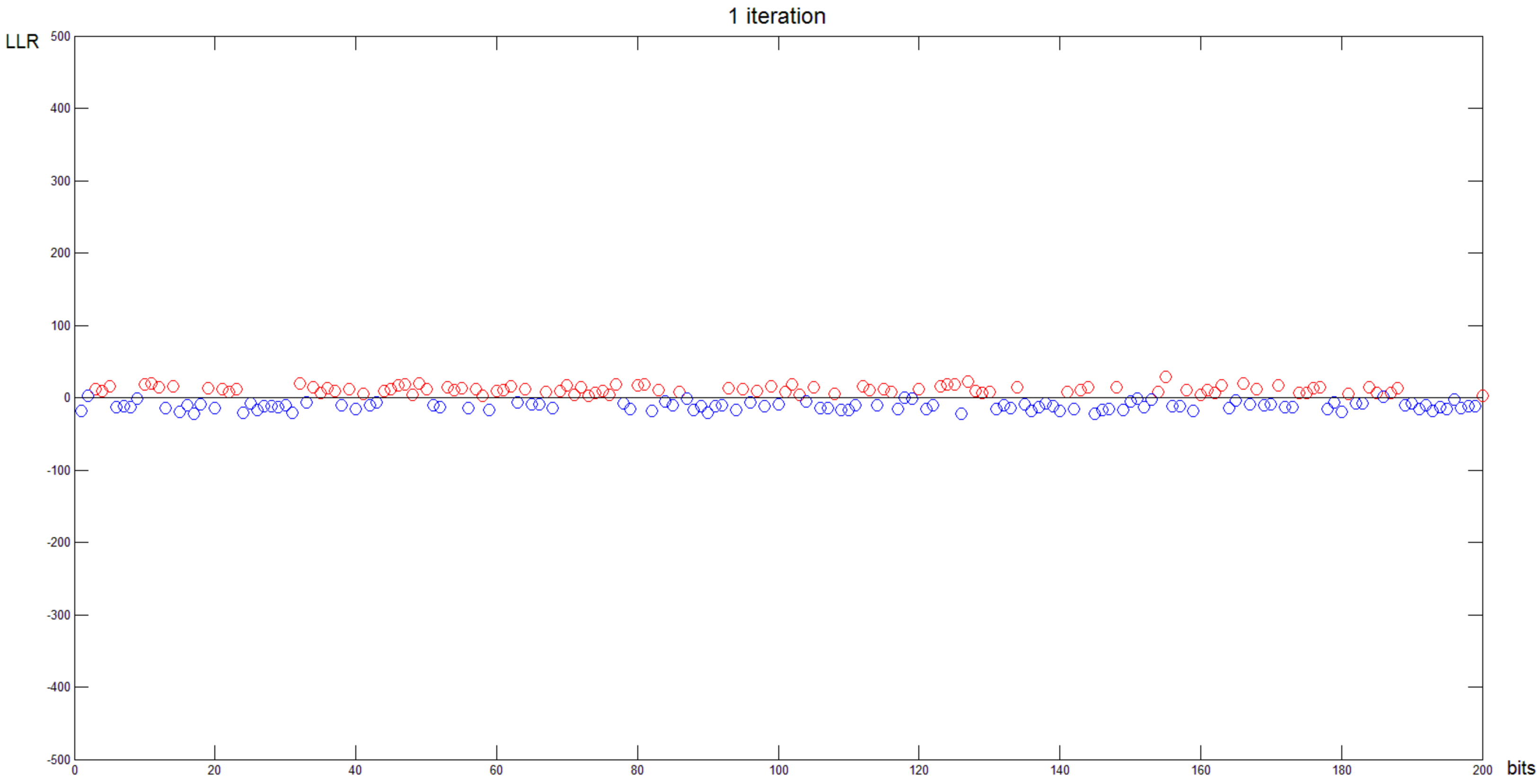
- WiFi 802.11n Standard
- Codeword size: 1944 bits
- Code Rate: 5/6
- Iteration: 0, 1 ,5, 10, 15
- SNR: 12 dB (Peak-to-peak/rms)

0 Iteration

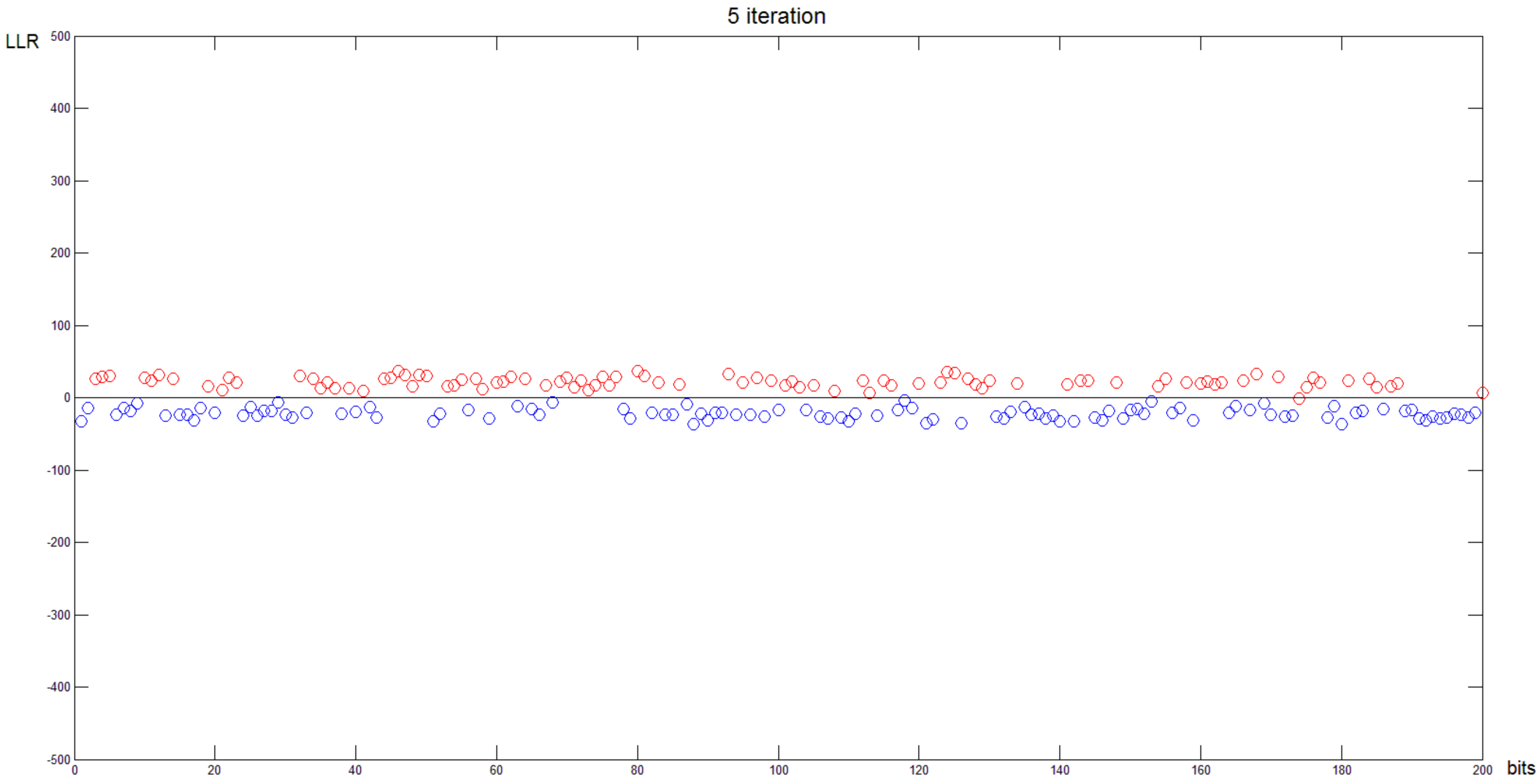


0 iteration
LLR
500
400
300
200
100
0
-100
-200
-300
-400
-500
0
20
40
60
80
100
120
140
160
180
200
bits

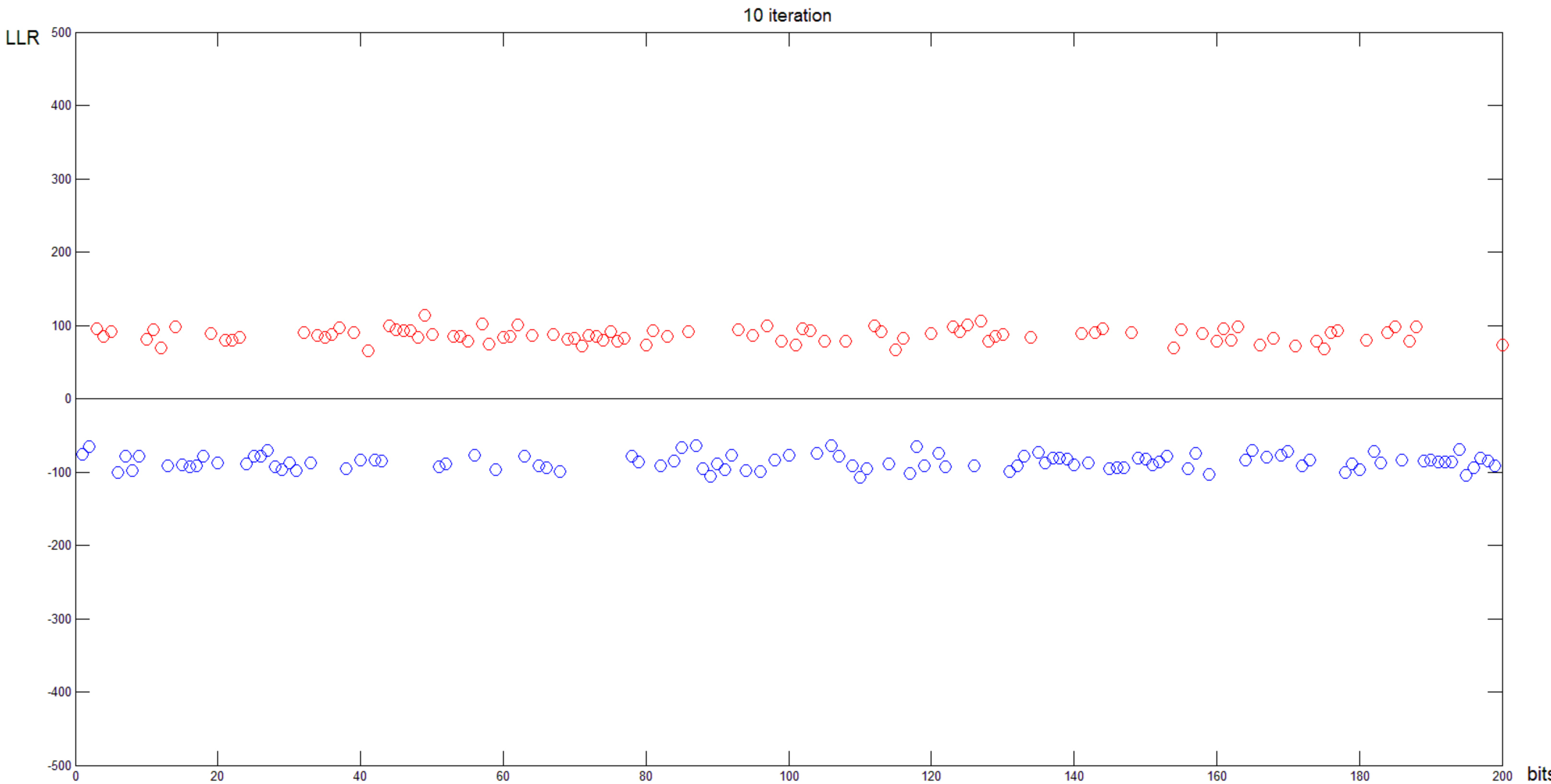
After 1 Iteration



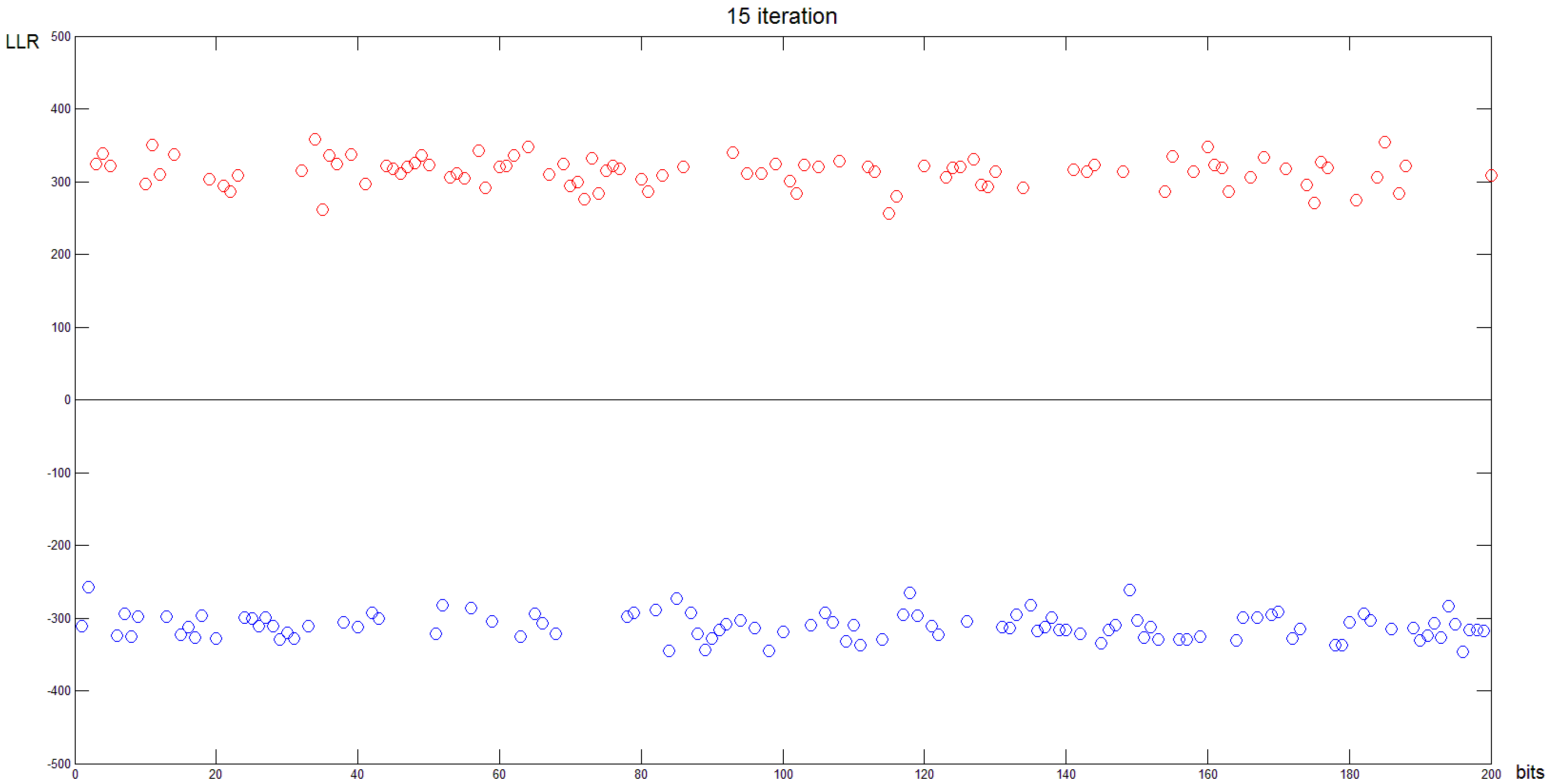
After 5 Iteration



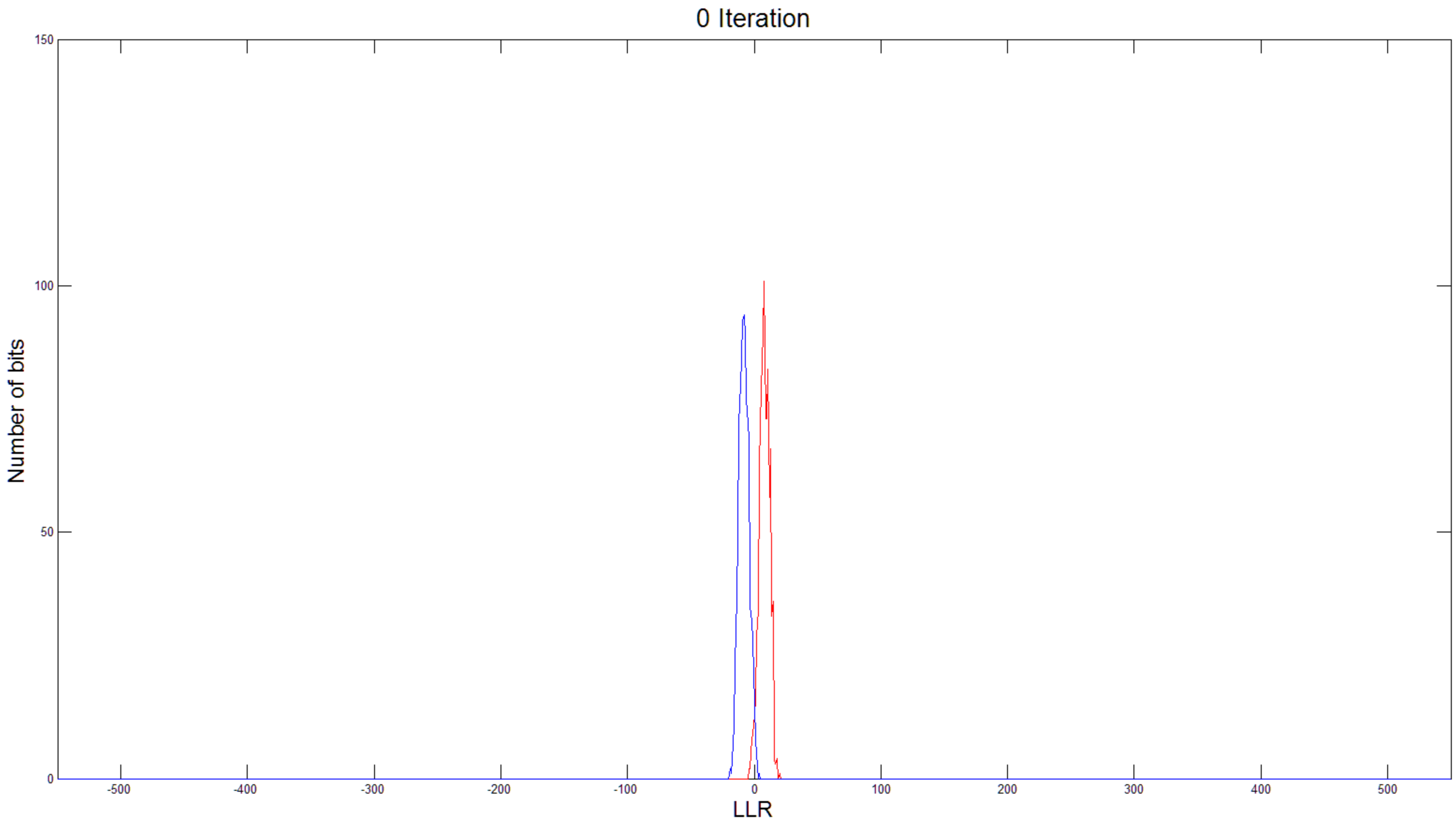
After 10 Iteration



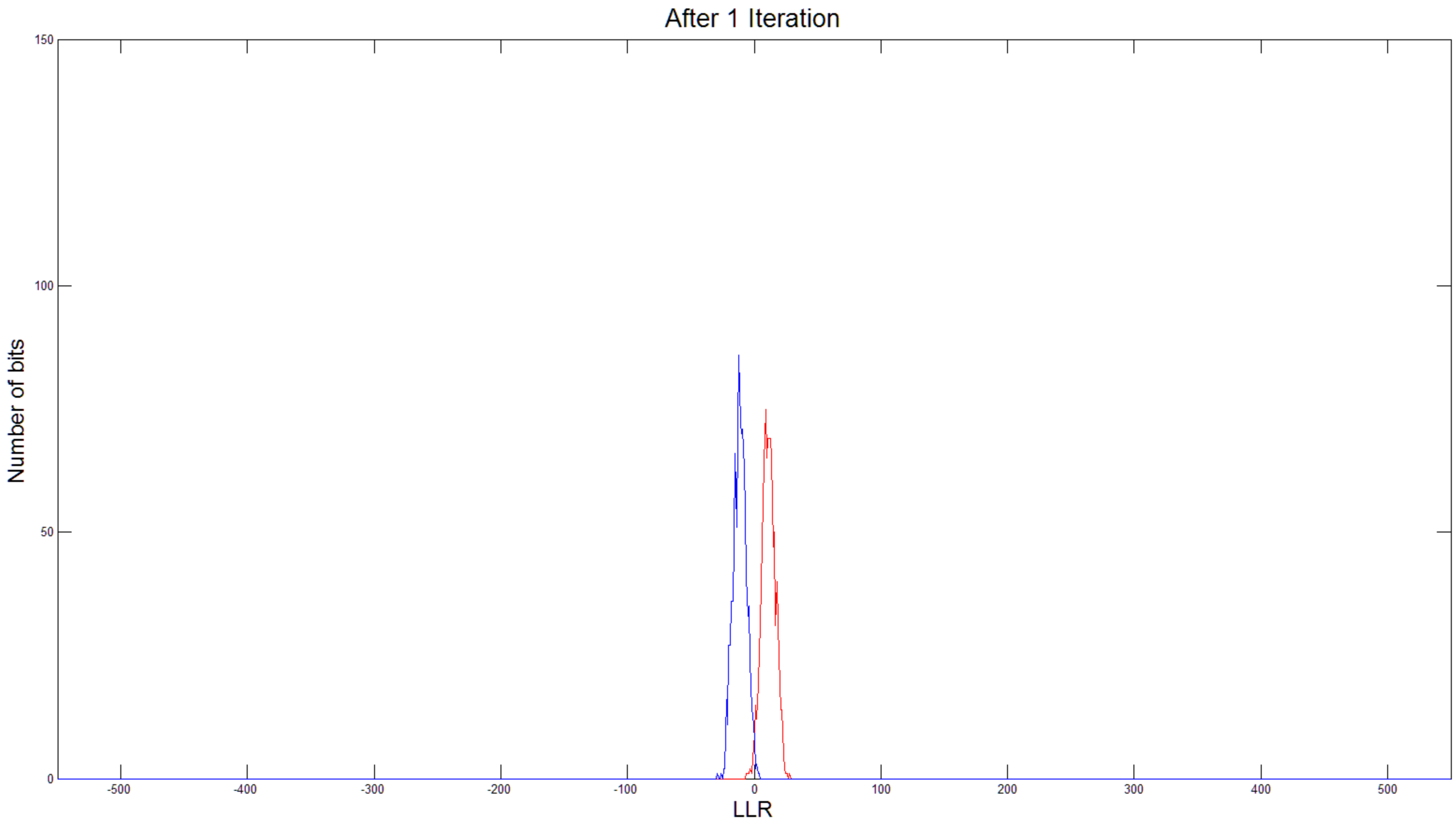
After 15 Iteration



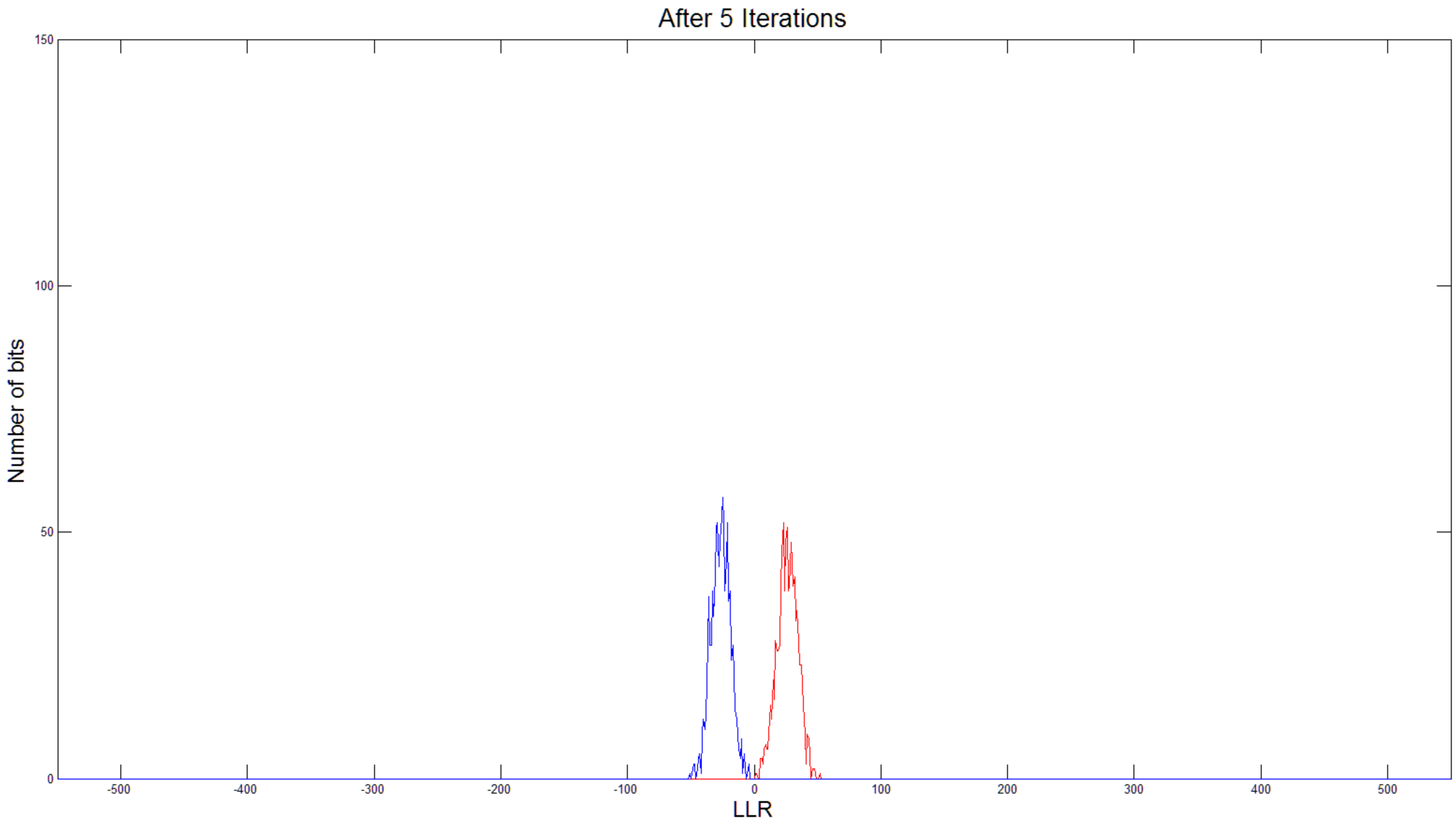
Histogram - 0 Iteration



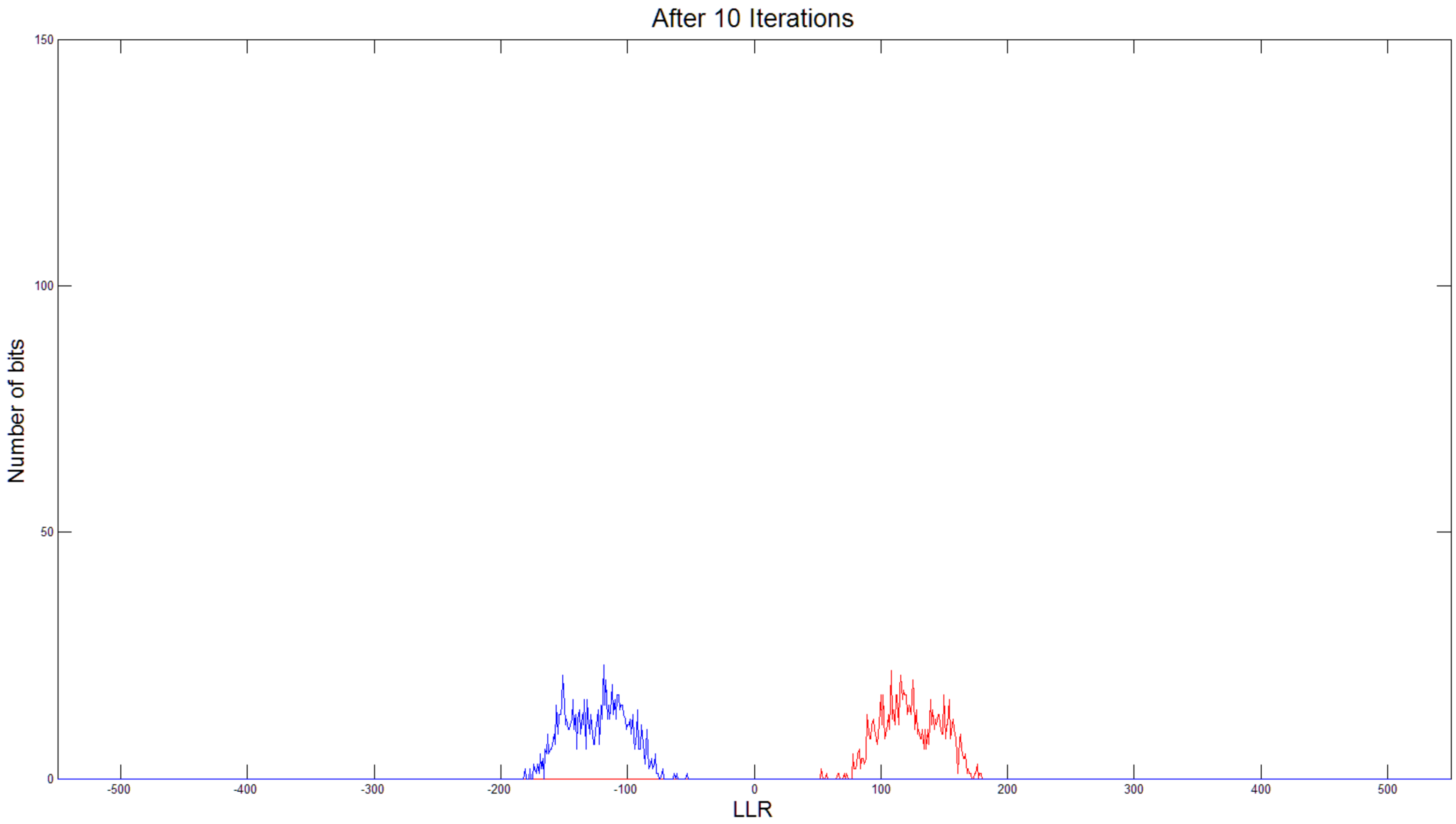
Histogram - After 1 iteration



Histogram – After 5 iterations

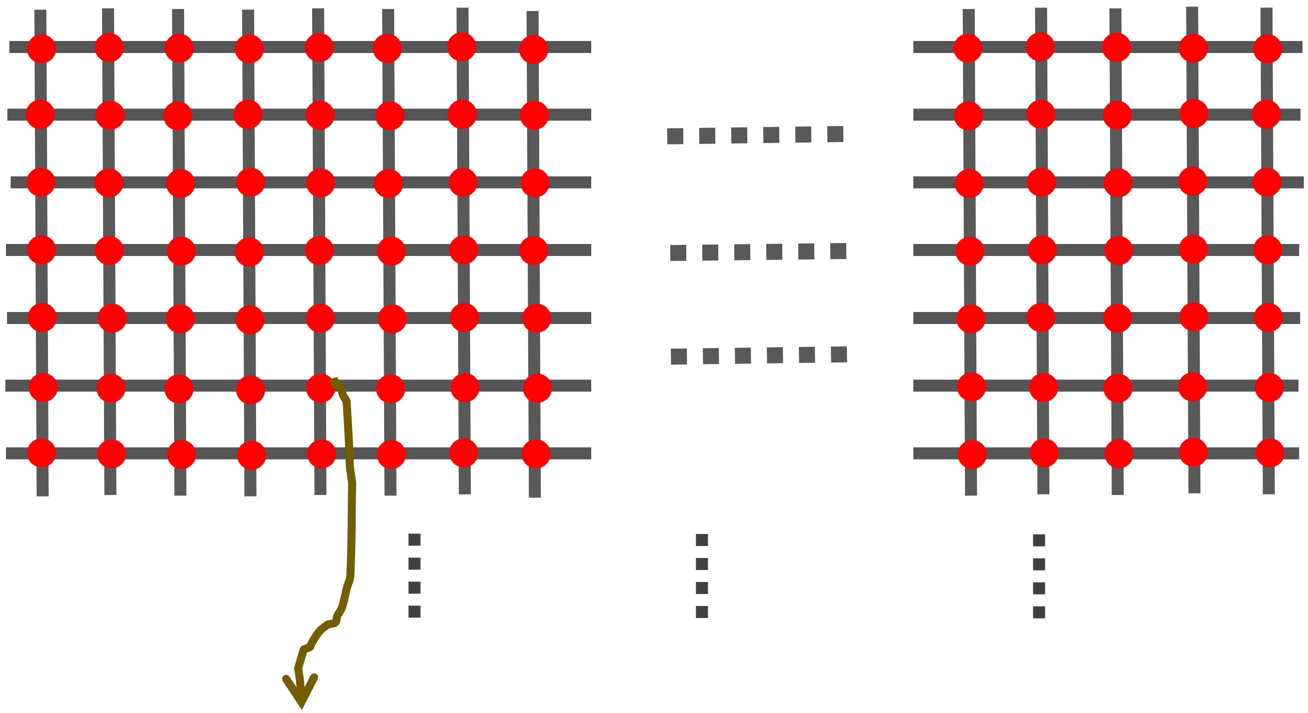


Histogram – After 10 iterations



Channel Modeling: Signal-Level Characterization of Cell Disturb

Channel Modeling: Cell Correlation



No cell correlation $r_{ij} = x_{ij} + n_{ij}$

Cell correlation through program/read disturb $r_{ij} = x_{ij} + \Delta(x_{ij}, \underbrace{x_{i_1 j_1}, x_{i_2 j_2}, x_{i_3 j_3}, x_{i_4 j_4}}_{\text{affecting cells}}) + n_k$

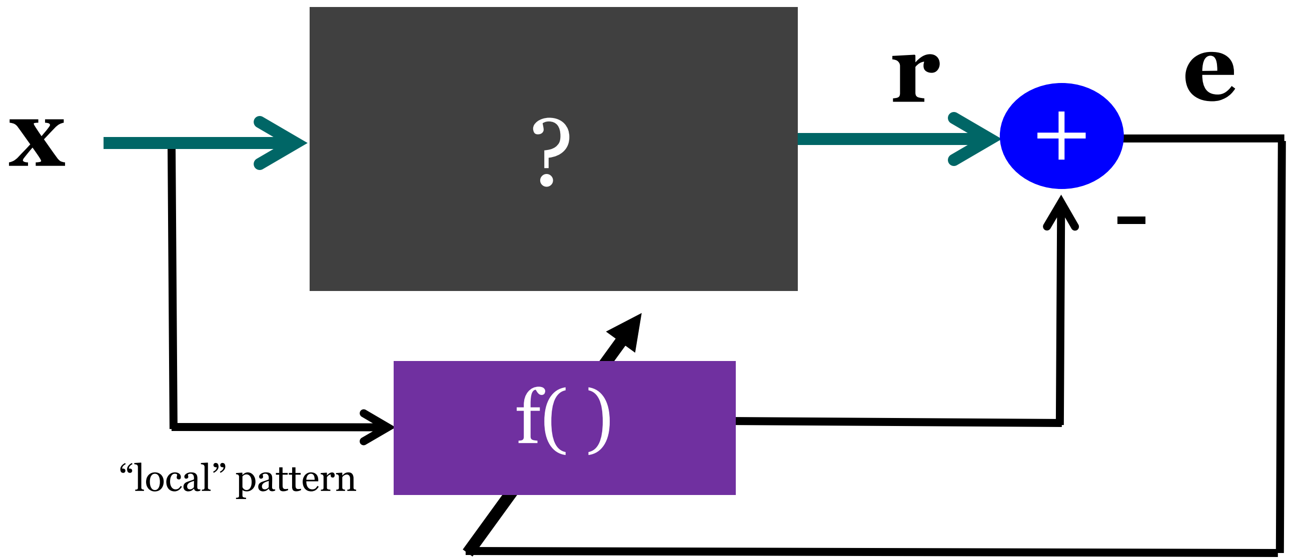
x_{ij}: victim cell

Channel Identification Problem

Feed the system with known data **x**.
Observe **r**.

Characterize the system enough, so for new data **x'** we would know what **r'** is.

Channel Identification Problem

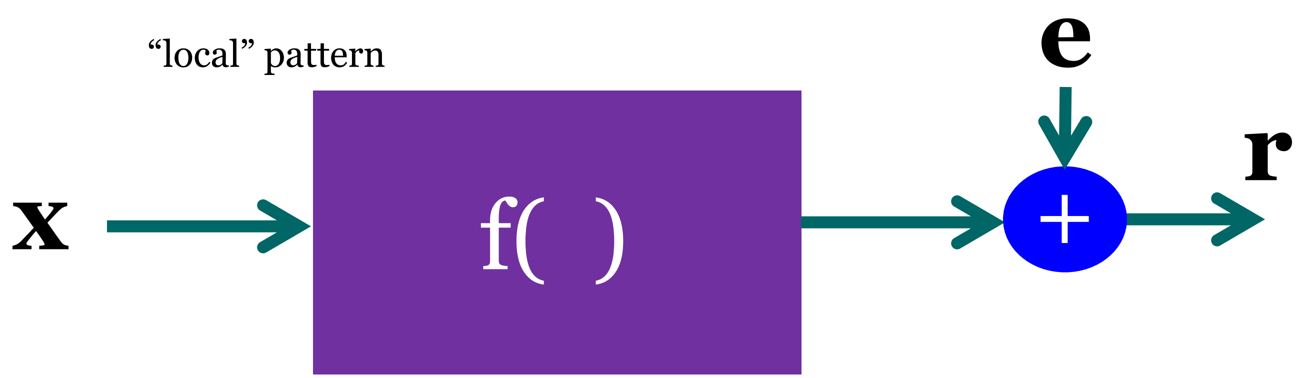


Feed the system with known data **x**.
Adjust f( ) until **e** is minimized (a sequential update algorithm is used).

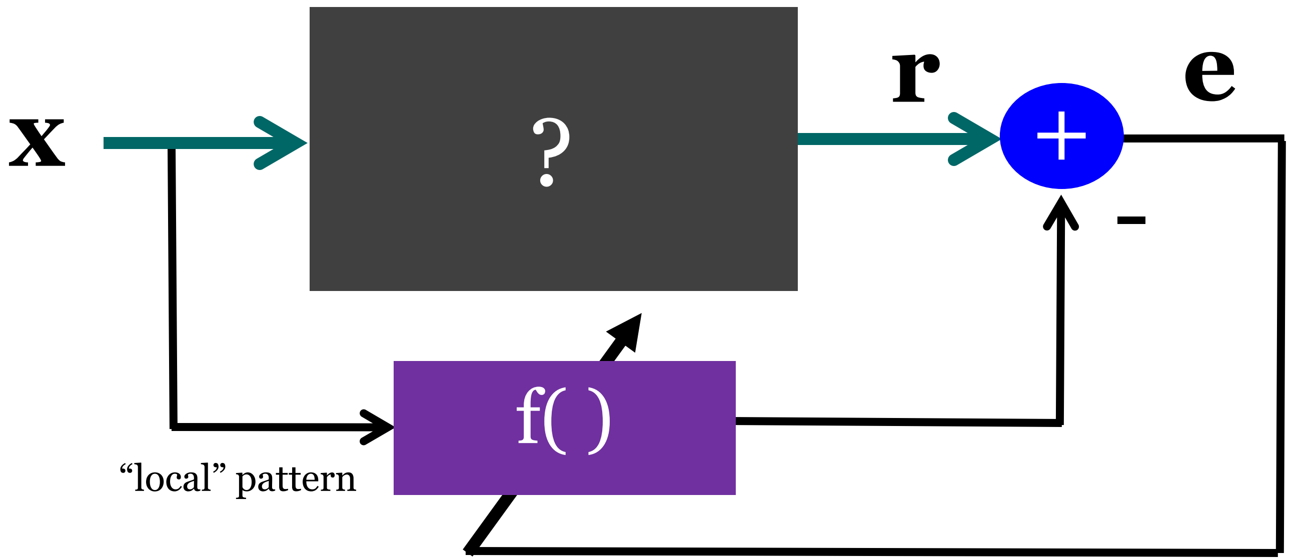
Once **e** stabilizes, f( ) should resemble the system closely.

We in essence are fitting the unknown box with a fixed model (with a know structure but with unknown parameters)

Characterized Channel



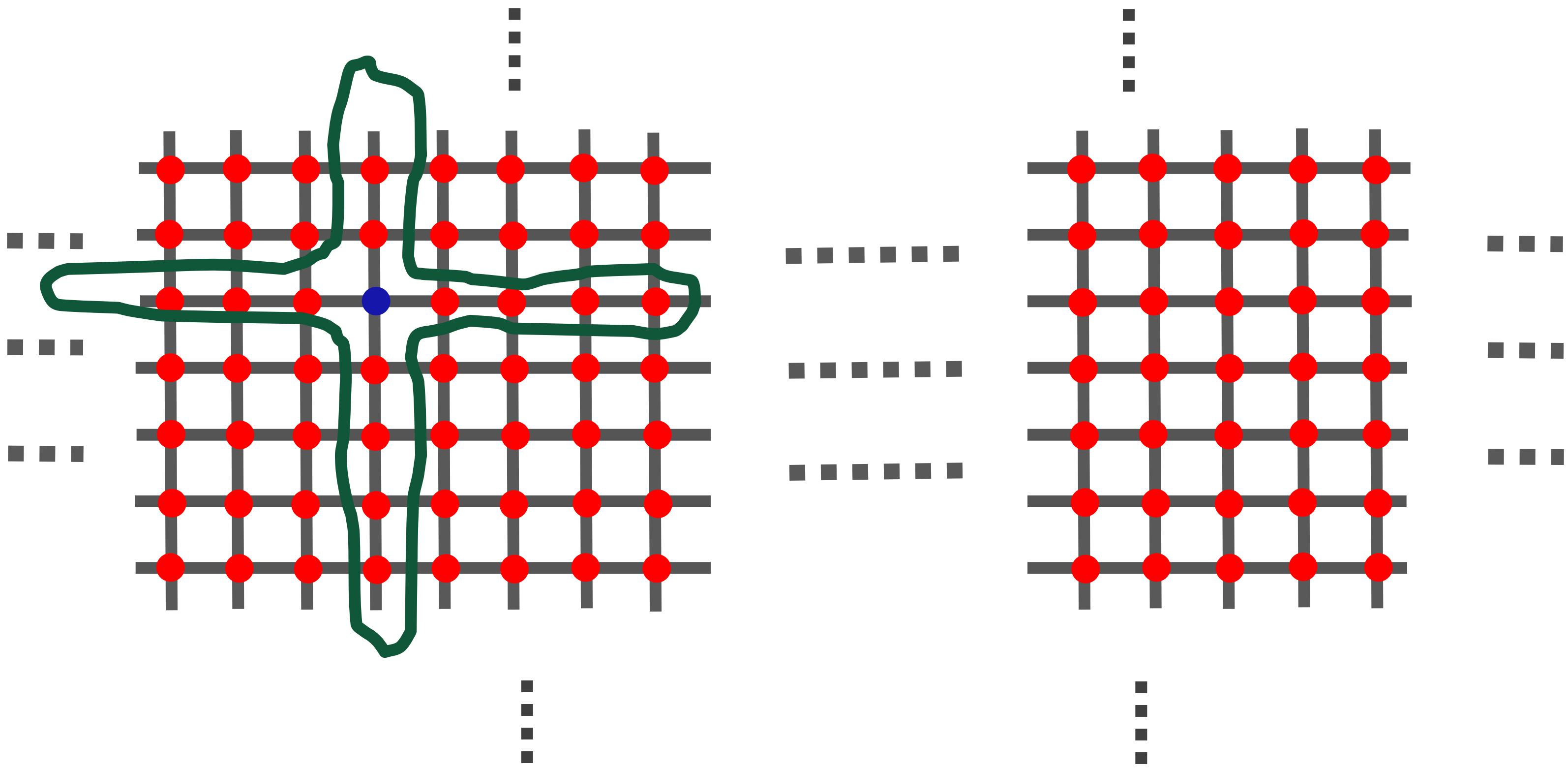
A general f( ): RAM



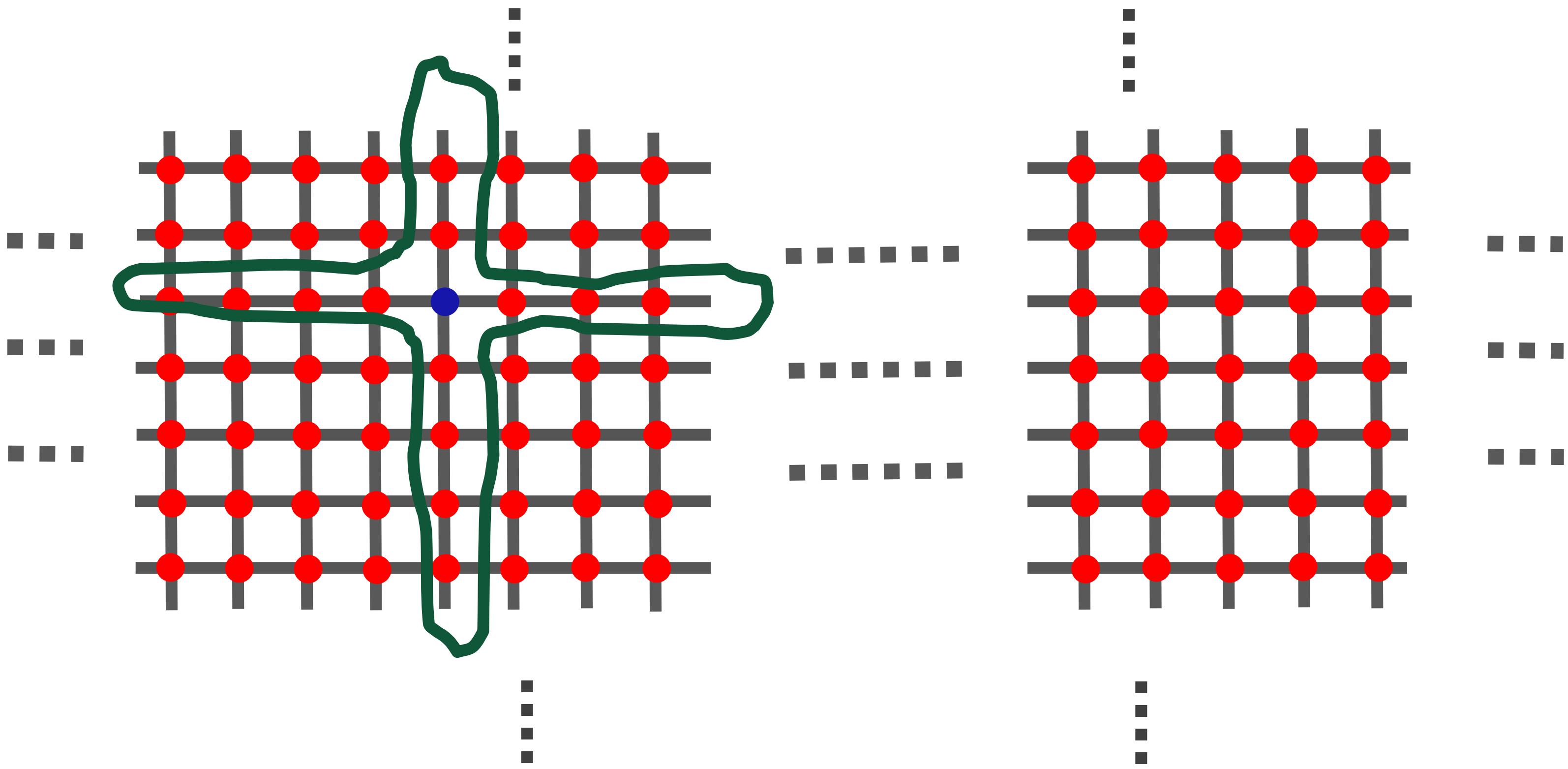
f( ) is a RAM and its contents are the cells' read values.
The local patterns of affecting cells are the RAM addresses.

When the error **e** is minimized (and stabilized), the RAM contents reflect the actual cell read values.

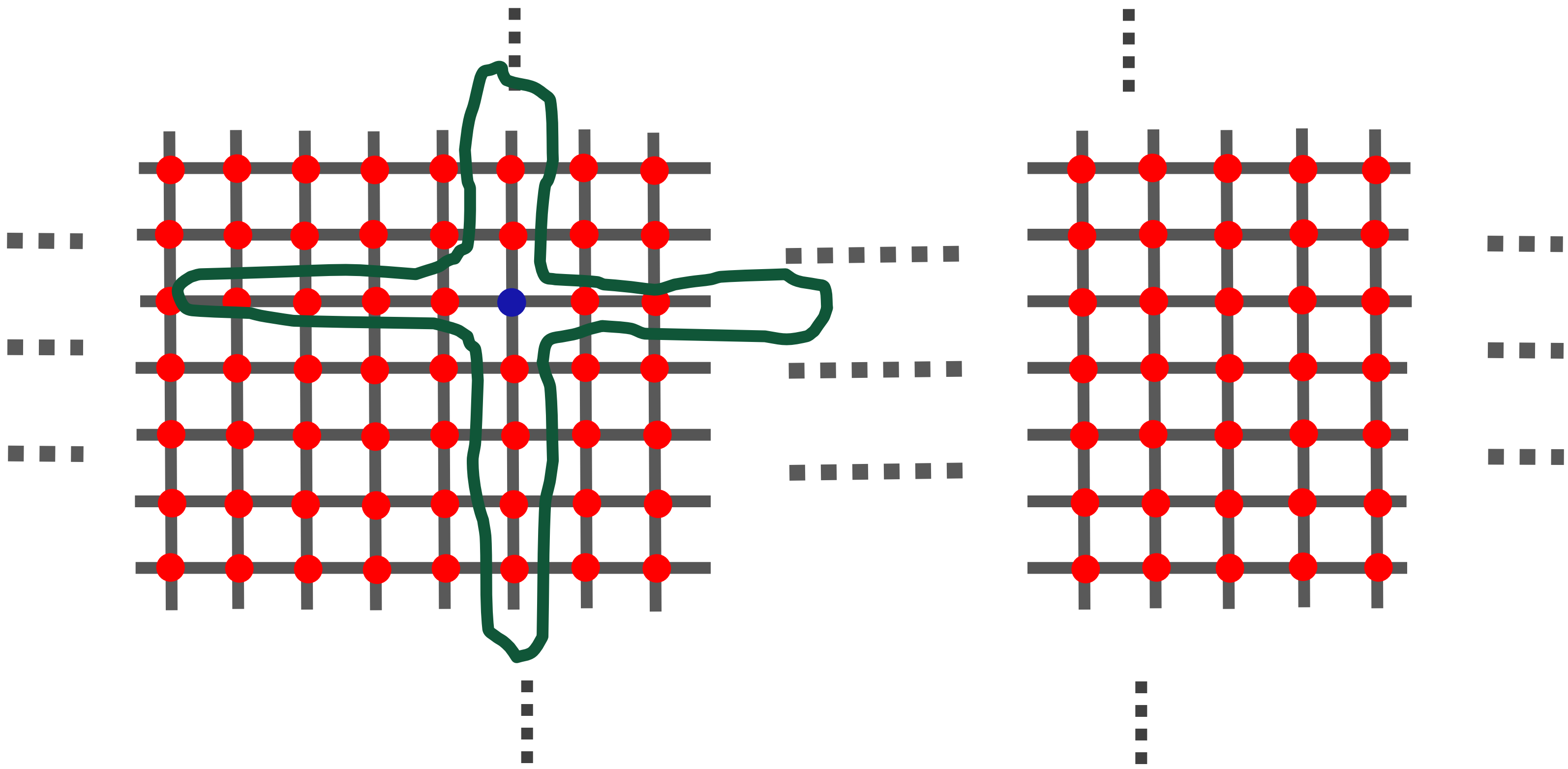
RAM Update Process: An Example of "Local" Pattern



RAM Update Process: An Example of "Local" Pattern



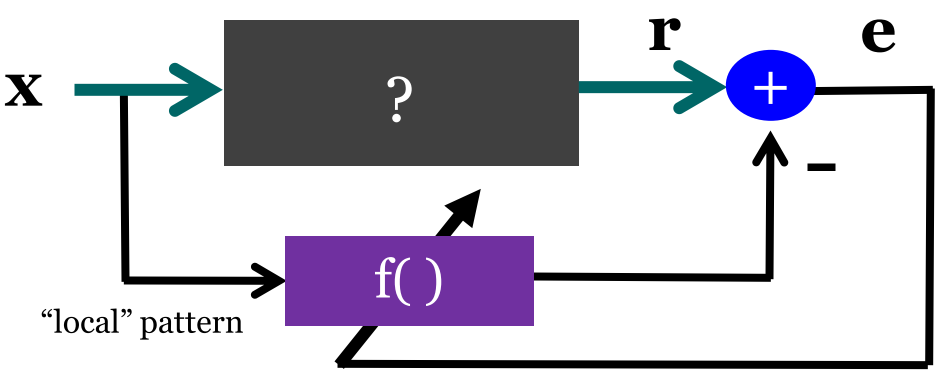
RAM Update Process: An Example of "Local" Pattern



RAM Update Process

$$f\Big(\overbrace{\underset{\text{victim cell}}{x_{ij}}\ ,\underbrace{x_{i,j-1}, x_{i-1,j}, x_{i,j+1}, x_{i+1,j}}_{\text{affecting cells}}}^{\textit{address}\ \text{(surrounding pattern)}}\Big) \approx r_{ij}$$

add1	add2	add3	add4
add5	add6		



$$f(adress1) \leftarrow f(adress1) + \mu \cdot r(address1) \cdot e(address1)$$

Do this for all addresses.

With Channel iteration