

# **Design of LDPC Decoders for Low Error Rate Performance**

Zhengya Zhang, Borivoje Nikolić  
Lara Dolecek, Venkat Anantharam, Martin Wainwright

Department of Electrical Engineering and Computer Sciences  
University of California, Berkeley

# Outline

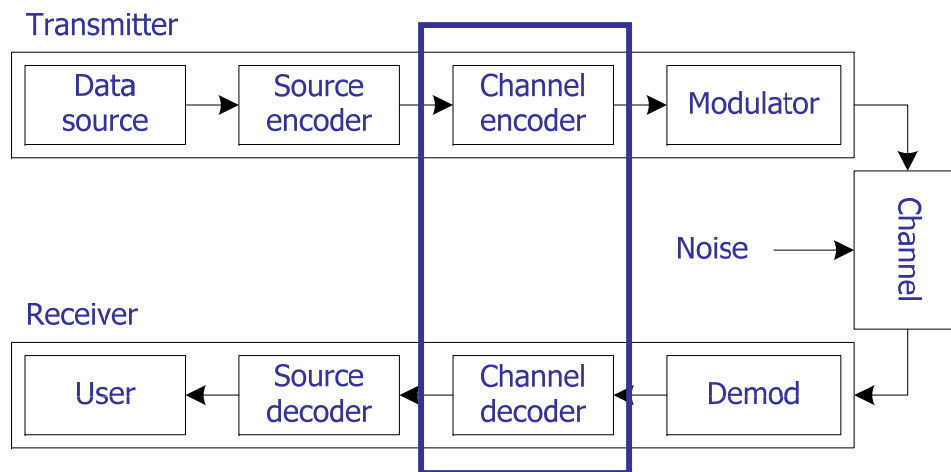
- LDPC code and iterative decoding
- An emulation-simulation approach to investigate the error floor of LDPC codes
- Improved implementations to lower the error floor

# Outline

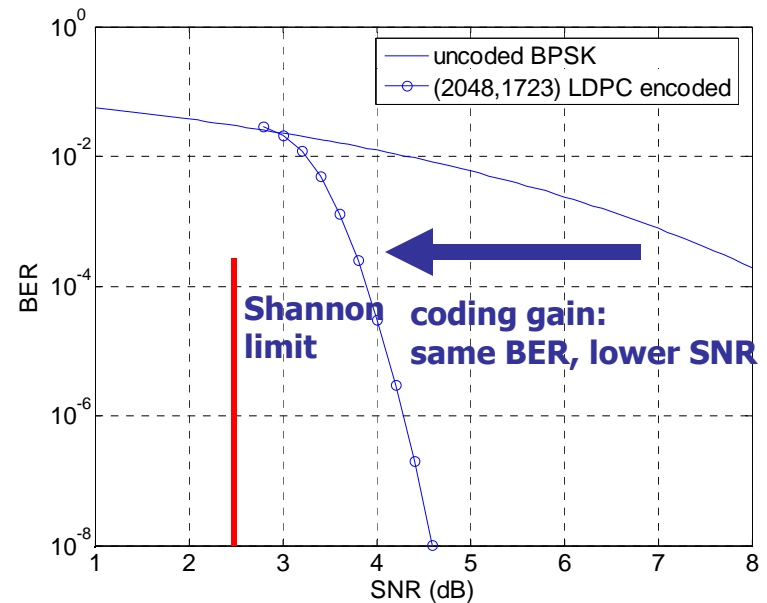
- LDPC code and iterative decoding
- An emulation-simulation approach to investigate the error floor of LDPC codes
- Improved implementations to lower the error floor

# Error Control Strategy

- Apply FEC to improve the reliability of transmission
- Low-density parity-check code has been demonstrated to come closest to the Shannon limit <sup>[1]</sup>



A communication system with forward error correction.



Uncoded and coded system performance comparison in an AWGN channel.

[1] Richardson, Shokrollahi, and Urbanke, *Trans. IT*, Feb. 2001.

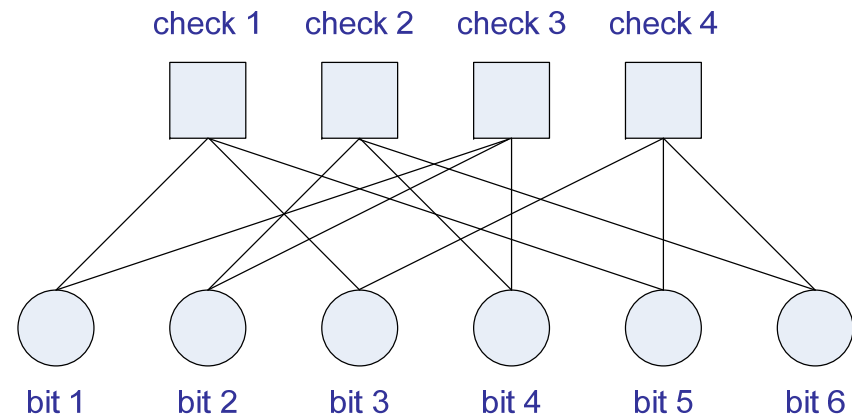
# Low-Density Parity-Check Code

Add redundancy to the transmitted data so that a block of  $N$  bits satisfy a set of  $M$  parity check equations.

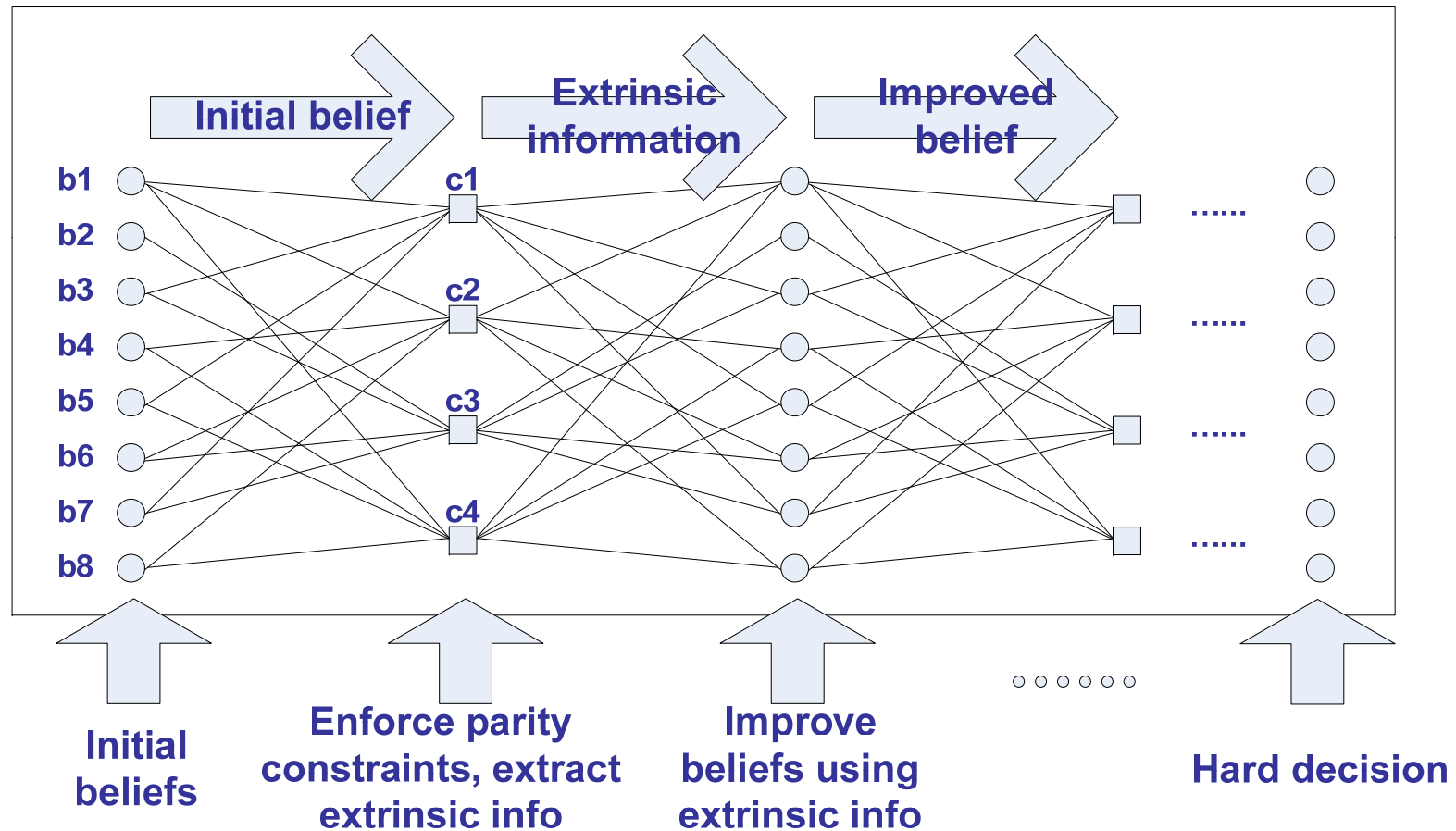
$M \times N$  parity check matrix (H matrix)

	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6
check 1	1	0	1	0	1	0
check 2	0	1	0	1	0	1
check 3	1	1	0	1	0	0
check 4	0	0	1	0	1	1

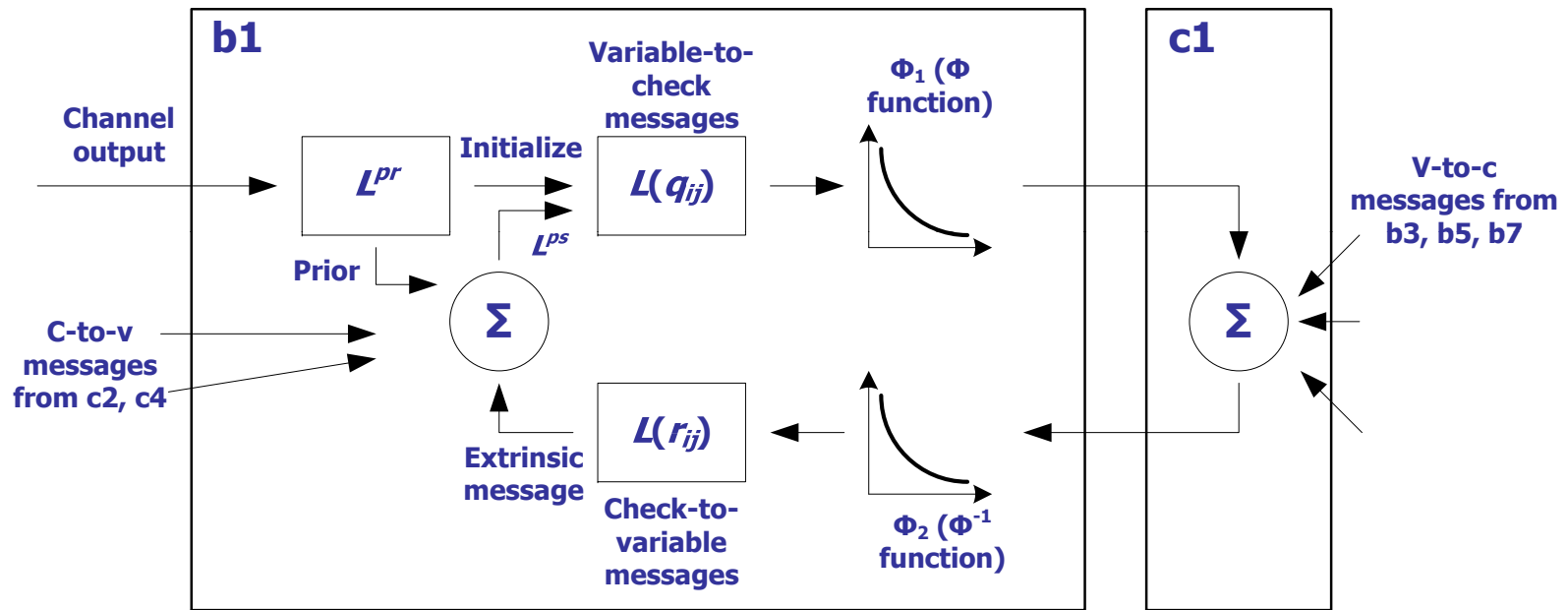
Tanner graph with  $M$  check nodes and  $N$  bit nodes



# Unfolding the Tanner Graph for Illustration



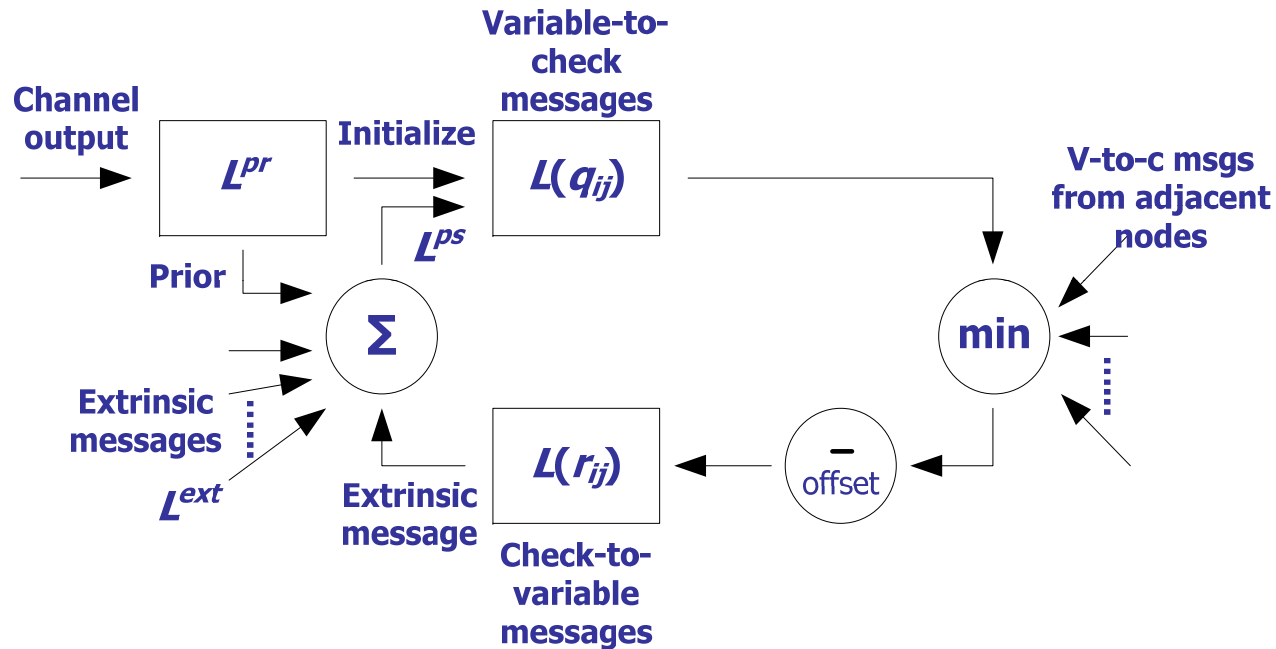
# Folding the Tanner Graph in Implementation



Showing one slice of the Tanner graph:

- 1 variable node and 1 check node
- Messages iterate between variable node and check node

# Min-Sum Approximation

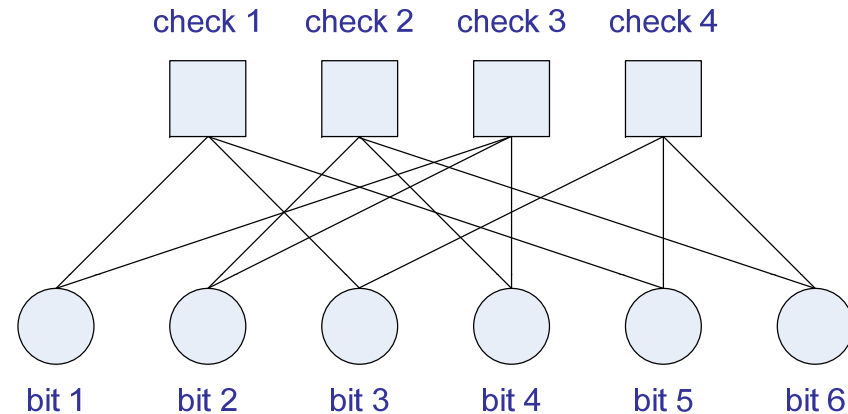


- Log-tanh functions can be costly to implement
- One popular approximation is to replace log-tanh functions with a min function
- The estimation error can be reduced by an offset correction or scaling <sup>[1]</sup>

[1] Chen, Dholakia, Eleftheriou, Fossorier, and Hu, *Trans. Comm.*, Aug. 2005.

# Message Passing Schedules and Decoder Architectures

	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6
check 1	1	0	1	0	1	0
check 2	0	1	0	1	0	1
check 3	1	1	0	1	0	0
check 4	0	0	1	0	1	1



## Fully parallel schedule/architecture [1]

- All VNs send messages to neighboring CNs concurrently
- Followed by all CNs send messages to neighboring VNs concurrently

## Fully serial schedule/architecture

- $b1 \rightarrow c1, b3 \rightarrow c1, b5 \rightarrow c1, b2 \rightarrow c2, \dots$

## Partially parallel schedule/architecture [2], [3]

- First subiteration:  $b1 \rightarrow c1, b3 \rightarrow c1, b5 \rightarrow c1$  concurrently
- Second subiteration:  $b2 \rightarrow c2, b4 \rightarrow c2, b6 \rightarrow c2$  concurrently

...

[1] Blanksby and Howland, *JSSC*, Mar. 2005.

[2] Mansour and Shanbhag, *ISLPED*, Aug. 2002.

[3] Yeo, Nikolic, and Anantharam, *Comm. Mag.*, Aug. 2003.

# Outline

- LDPC code and iterative decoding
- An emulation-simulation approach to investigate the error floor of LDPC codes
- Improved implementations to lower the error floor

# Performance of LDPC Code

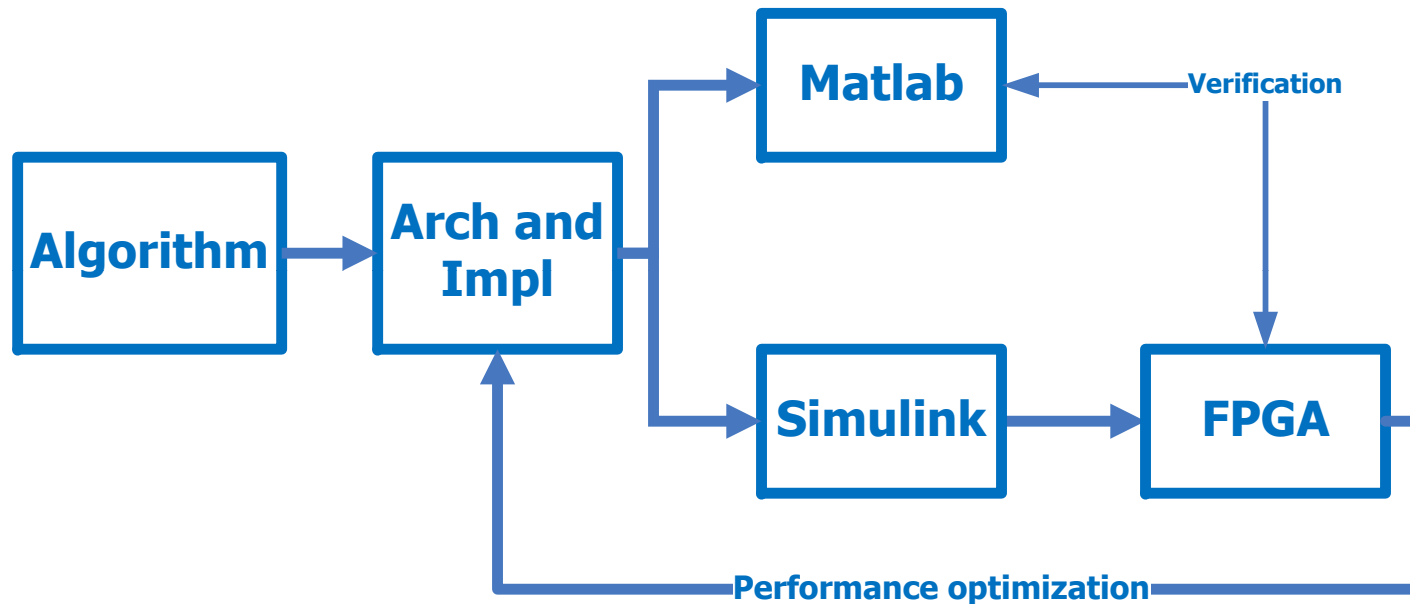
- Infinite block length
  - The performance of suitably designed codes of sufficiently large block length is known to be very close to Shannon limits <sup>[1]</sup>
  - Analytical techniques, such as density evolution <sup>[1]</sup> and EXIT charts <sup>[2]</sup>, have been developed for predicting the performance of iteratively decoded LDPC codes
  - These techniques are based on asymptotic approximations – very large block lengths, irregular construction, loop-free graph
- Moderate block length, high to moderate BER levels
  - Decoder performance can be easily simulated in software
  - Many code constructions (regular and irregular) with excellent waterfall performance have been proposed
- Moderate block length, low BER levels
  - Very important in practice, but no analytical approach, nor it can be easily simulated

---

[1] Richardson, Shokrollahi, and Urbanke, *Trans. IT*, Feb. 2001.

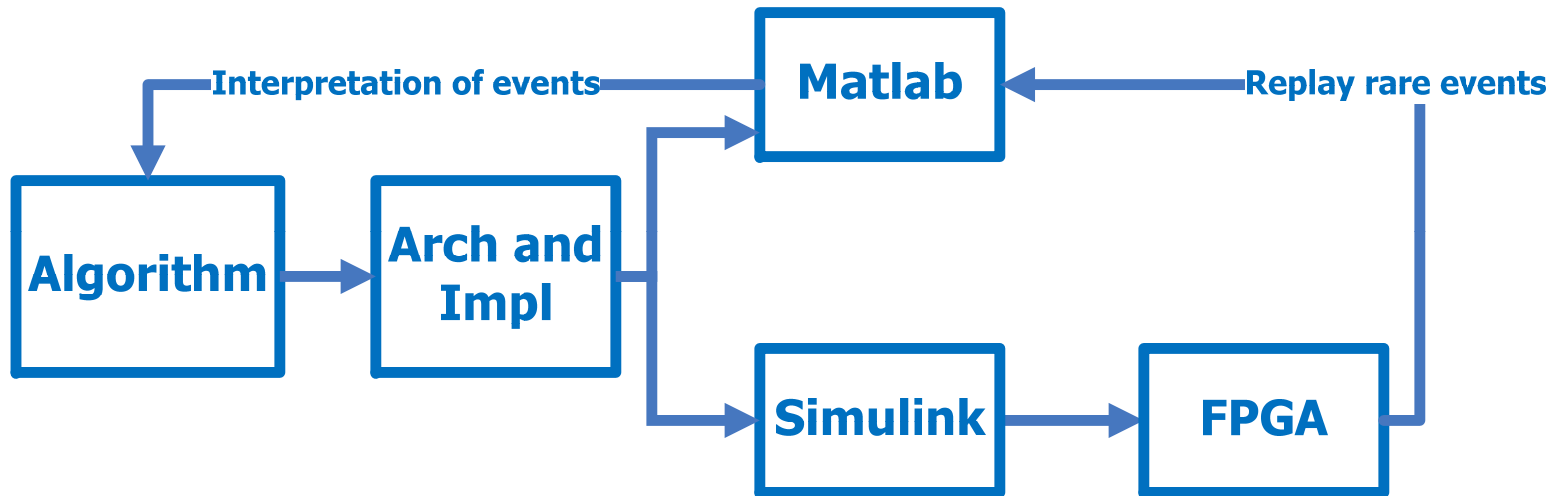
[2] ten Brink, *Electronic Letters*, May 1999.

# LDPC Decoder Emulation



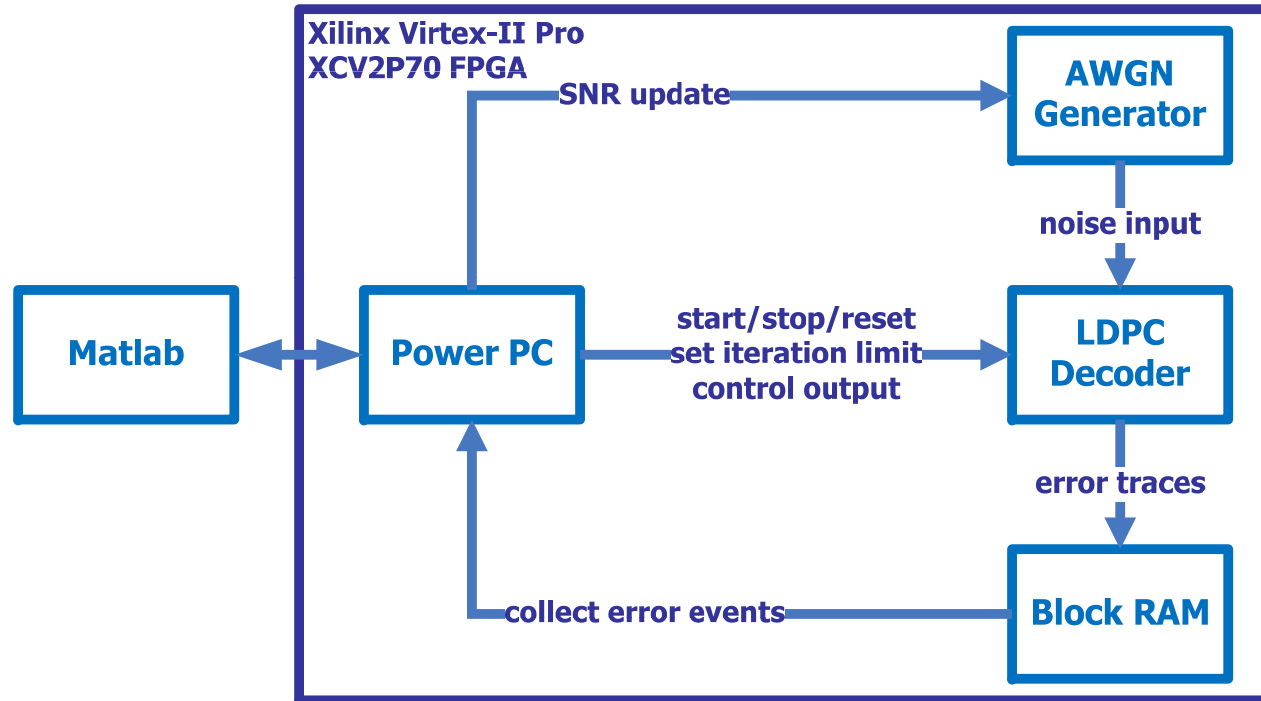
- A configurable FPGA emulation platform and a Simulink-based design flow for fast system prototyping
- High emulation throughput yields rare events (errors), that determinate the low BER performance of the code
- The usual approach: design by emulation – tune architecture and implementation for the best possible performance

# An Emulation-Simulation Approach



- Capture rare events (errors) on FPGA, which can be simulated (replayed) step-by-step in Matlab to reveal more insights
- The simulation results shed light on the causes of errors, when correlated with code structure and decoding algorithm, can yield in-depth results
- Analytical approaches can be developed to guide the optimization of both the algorithm and the implementation to improve performance

# Setup for Decoder Emulation

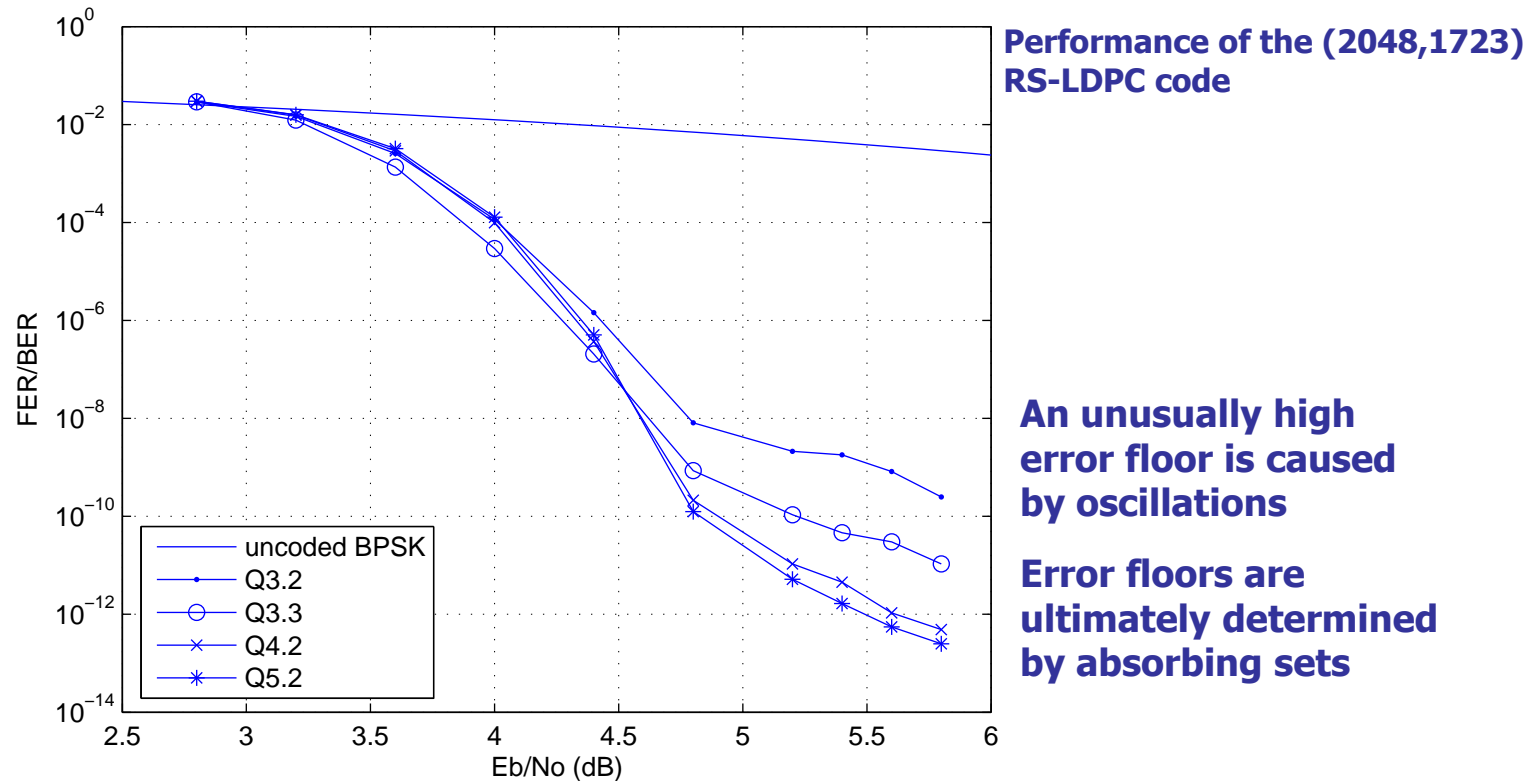


- Efficient decoder design to allow room on-chip to store error traces
- Configurable decoder architecture to facilitate the adaption to different codes, decoding algorithms, wordlength, and quantization

# Outline

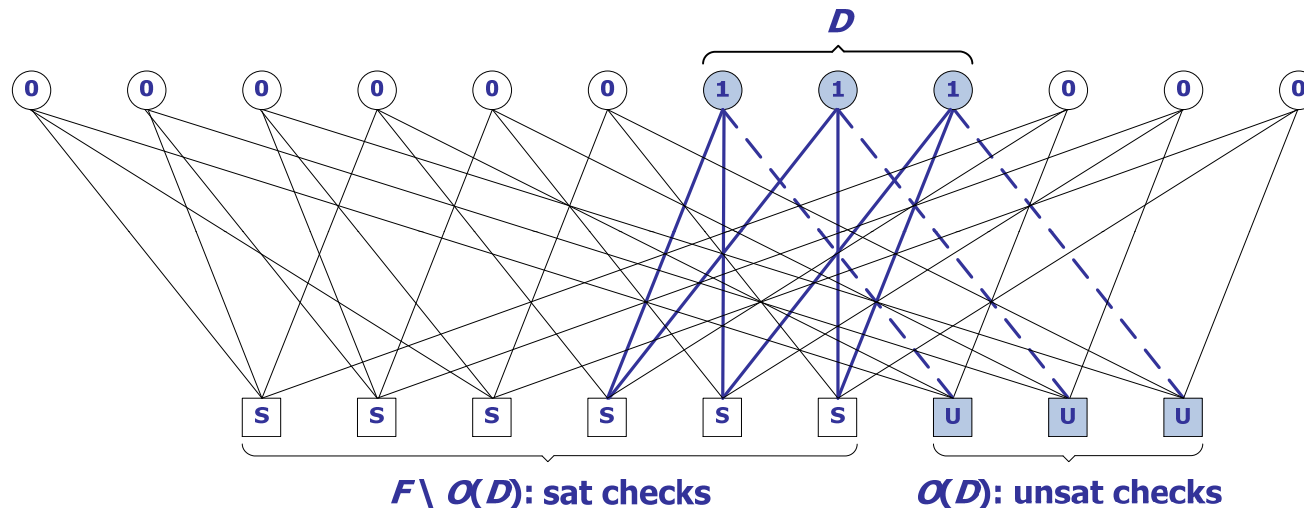
- LDPC code and iterative decoding
- An emulation-simulation approach to investigate the error floor of LDPC codes
- Improved implementations to lower the error floor

# Emulation of RS-LDPC SPA Decoder



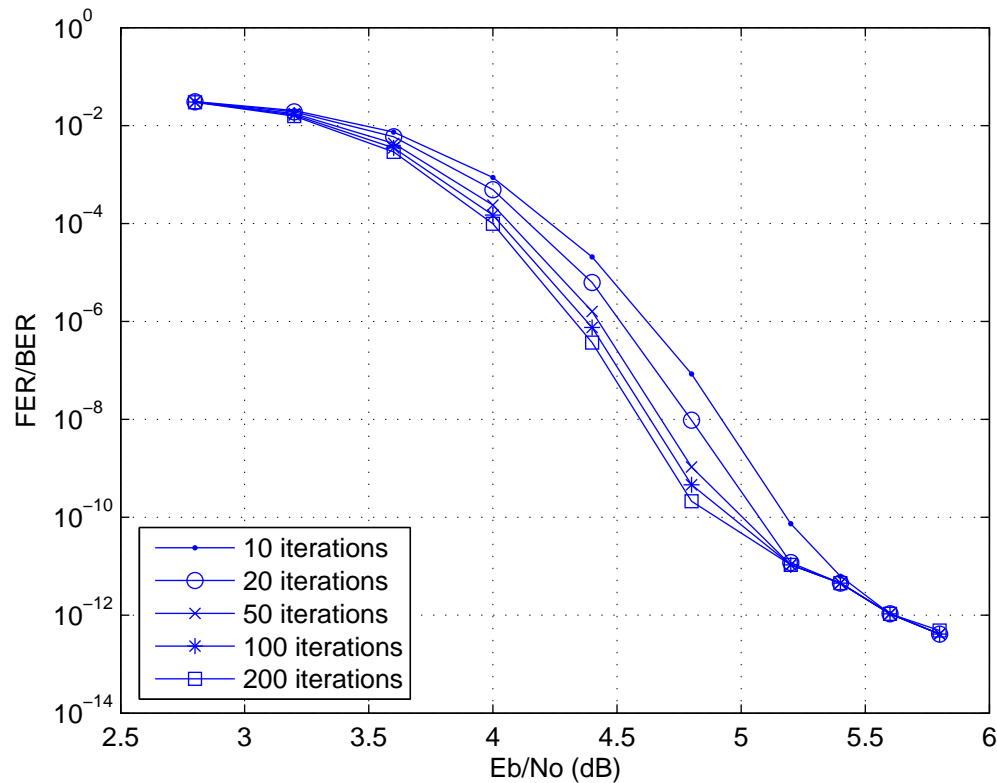
- Oscillatory behavior is induced by finite wordlength
- Absorbing sets are intrinsic to the code structure, but the effects of which can be exacerbated by quantization

# Absorbing Sets Illustrated



- Each variable node in  $D$  connects to more satisfied check nodes than the unsatisfied check nodes
- These nodes could never recover from the errors because they are reinforced by the satisfied check nodes which outnumber the unsatisfied check nodes
- The structure is stable under bit-flipping algorithm or a majority counting algorithm

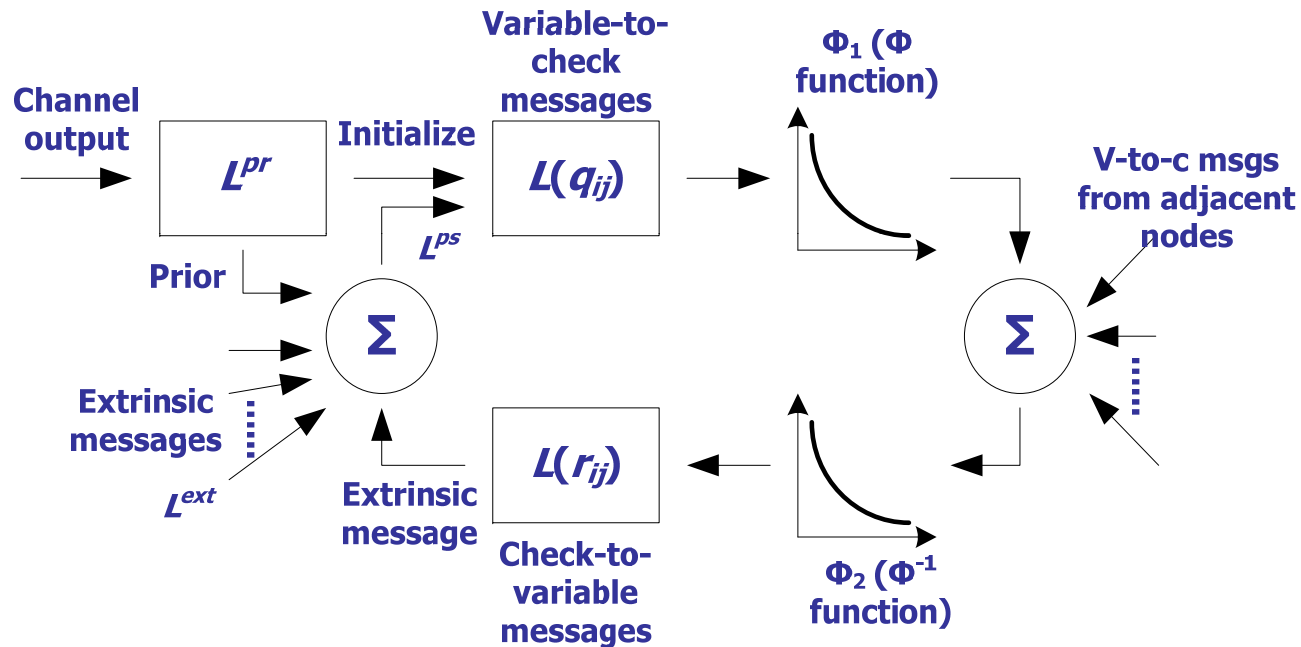
# Run More Decoding Iterations



- Error floors are due to the same absorbing sets
- In a high SNR level, absorbing process happens very quickly, usually within a few iterations
- Running more decoding iteration does NOT help alleviate absorbing states

Performance of the (2048,1723) RS-LDPC code

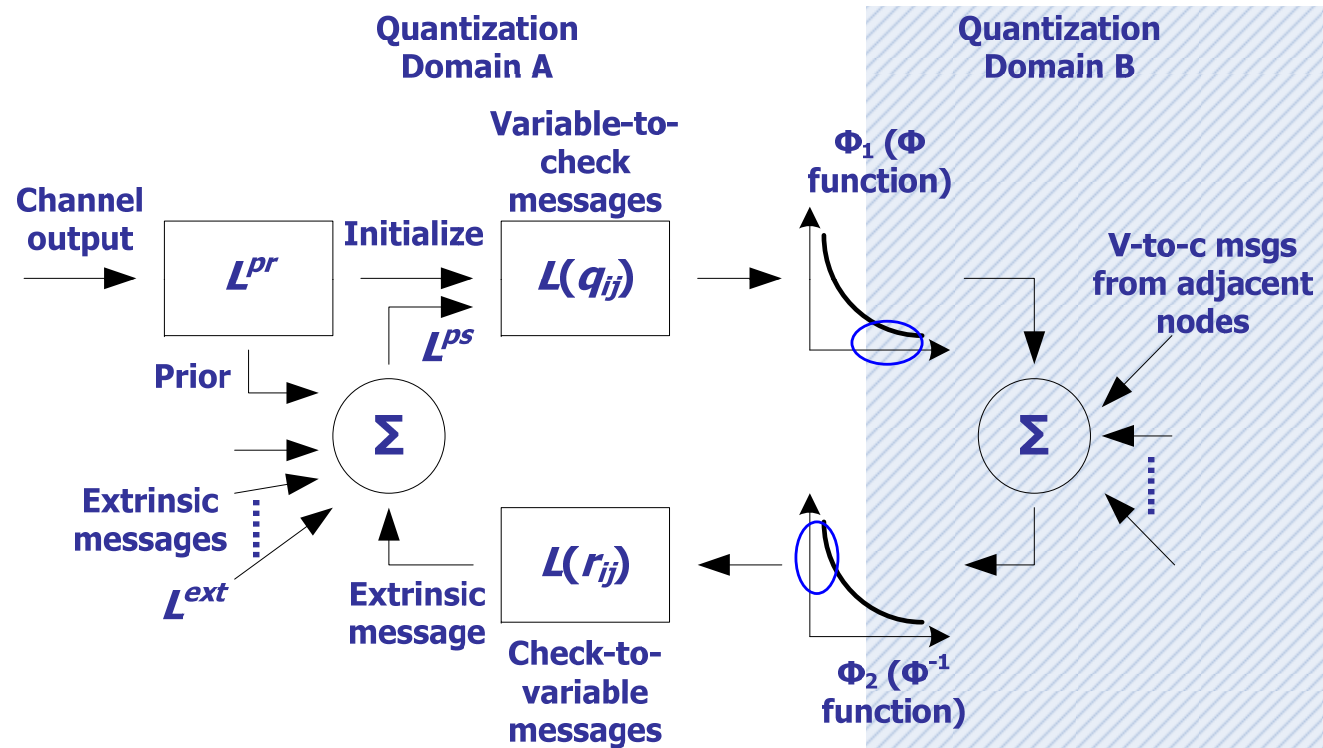
# Implementation-induced Weaknesses



The approximation of  $\log \tanh(\Phi)$  function causes overestimation of less reliable messages and underestimation of very reliable messages

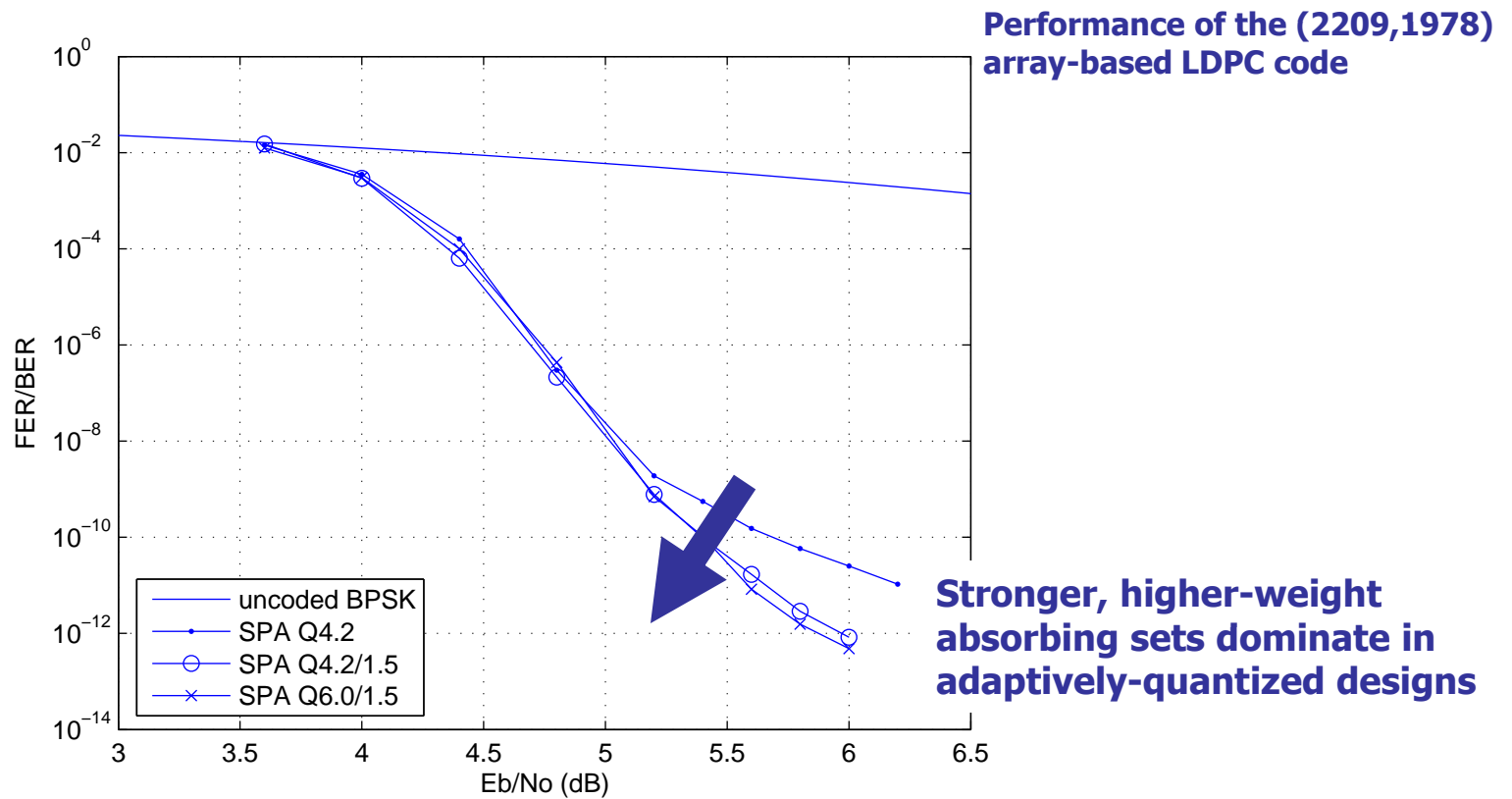
- Need to improve function estimation
- Or eliminate the log-tanh functions

# Improved Function Estimation

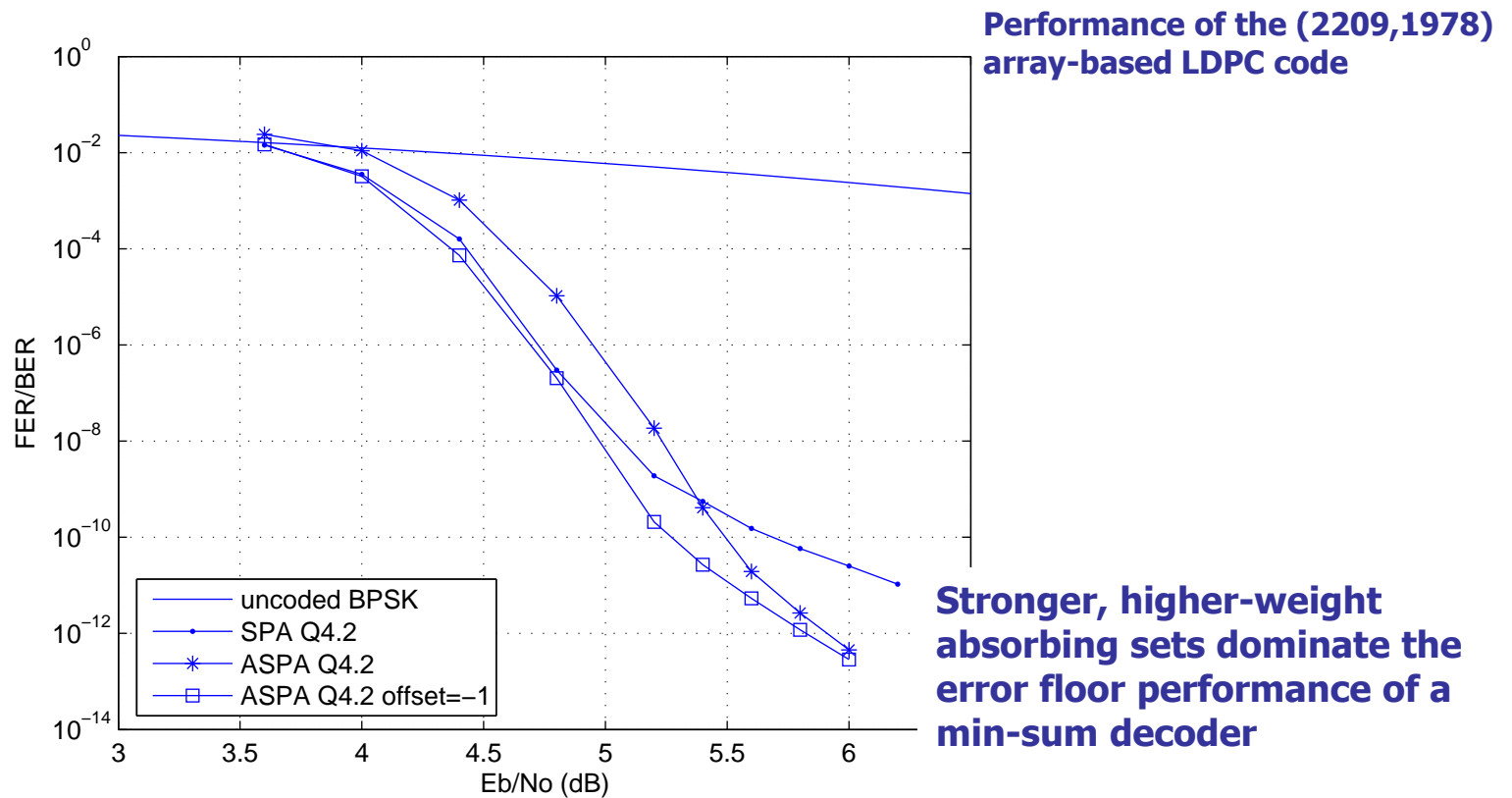


- In high SNR, messages tend to crowd towards the corners of the logtanh ( $\Phi$ ) function
- Divide the design into domains and quantization choices are tailored to the operation region within the domain

# Adaptive Quantization



# Eliminate Message Saturation with Min-Sum Approximation



# Conclusions

- Demonstrated an emulation-simulation approach in studying the low error rate performance of moderate block length, practical LDPC codes
- Determined the causes of error floors as absorbing sets, whose effects can be exacerbated by practical implementations
- Proposed improved implementation techniques to minimize effects of absorbing sets and lower the error floor

# Acknowledgement

- This research was supported in part by NSF CCF grant no. 0635372, Marvell Semiconductor, Intel Corporation, and Infineon Technologies through the UC MICRO program
- NSF CNS RI grant no. 0403427 provided the computing infrastructure
- Contributions of the students, faculty, and sponsors of Berkeley Wireless Research Center