Low-complexity Low-Precision LDPC Decoding for SSD Controllers

Shiva Planjery, David Declercq, and Bane Vasic

Codelucida, LLC

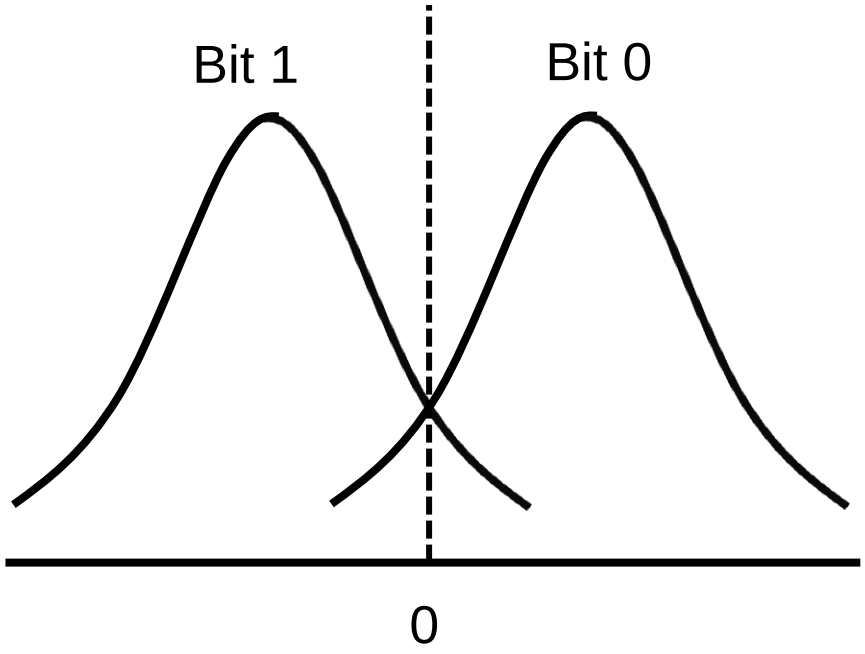
Website: www.codelucida.com

Email : planjery@codelucida.com

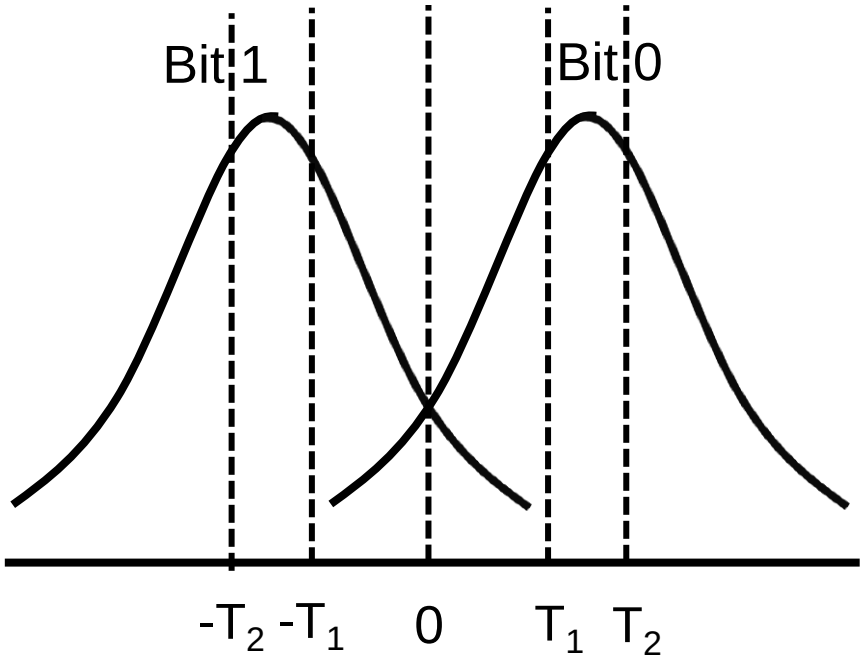
ECC Needs (LDPC vs BCH)

- With scaling of flash geometries, stronger ECC needed to enhance reliability especially for enabling MLC and TLC
- BCH codes provide guaranteed error-correction of *t* but operate with only hard-decision decoding.
- LDPC codes emerging as strong candidates as they can provide much higher reliability by using soft-information.

Hard vs Soft Information



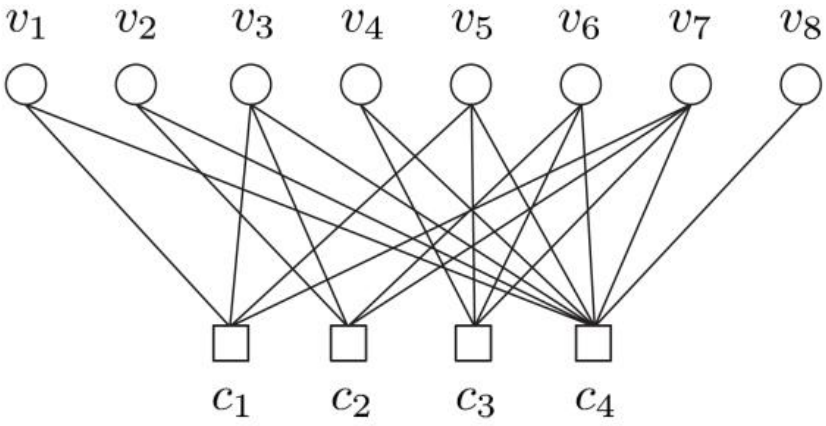
Hard-decision:1 read



Soft-decision: Multiple reads

Higher precision in soft-information requires more number of reads which add to the read latency

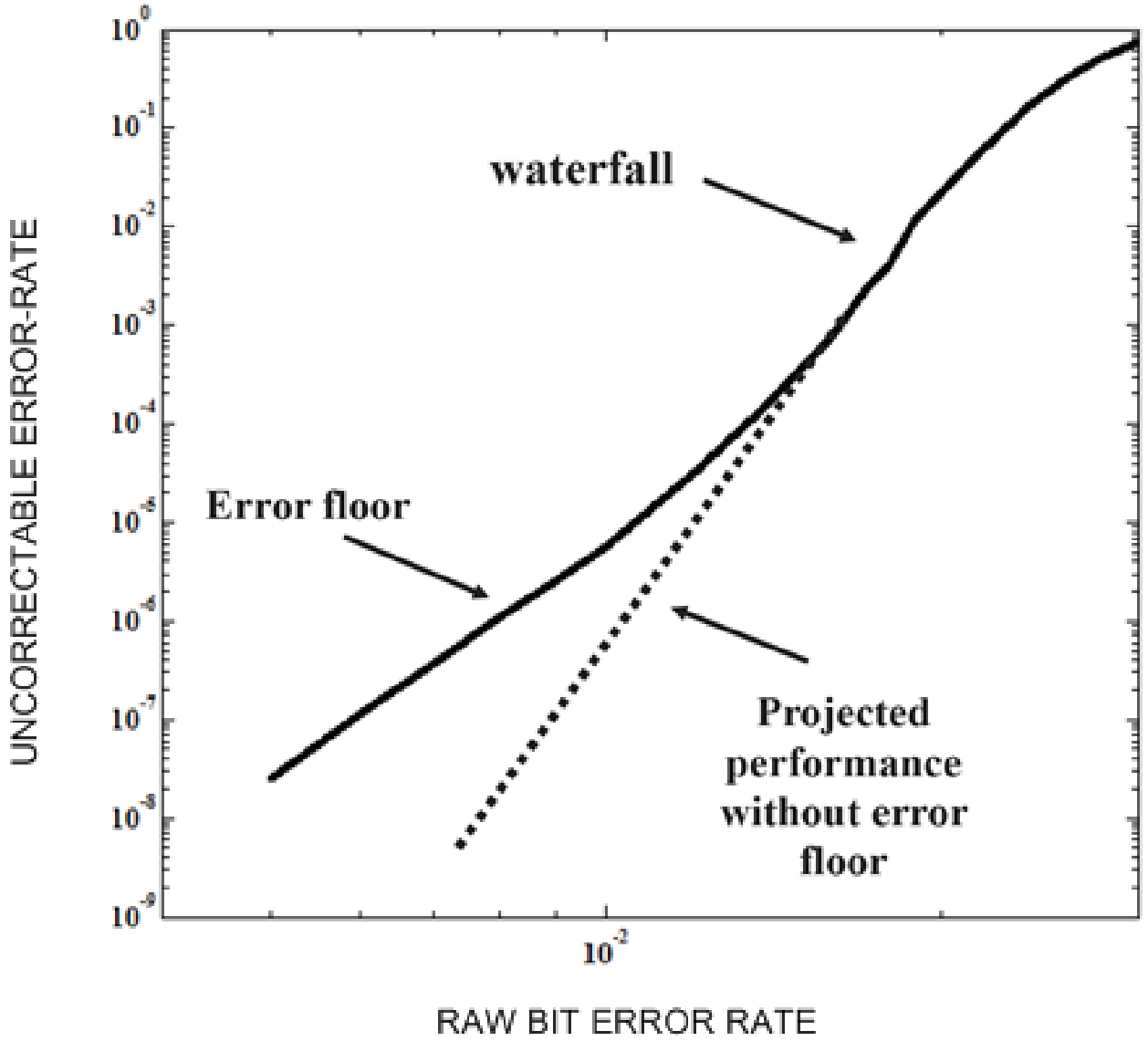
LDPC decoding: Message - Passing



Example of a Tanner graph

- Messages passed on the Tanner graph between variable nodes and checks nodes
- Messages are (quantized) log-likelihood ratios (LLRs)
- Each variable node receives an LLR from the channel referred to as c*hannel value*

Typical LDPC Performance



Lingering Issues and Challenges with LDPC designs for Flash

- Very high code-rates (6-10% redundancy).
- Design of good quasi-cyclic codes for code lengths 1KB, 2KB, 4KB and higher.
- Very low error-rate requirements (UBER $\approx 10^{-15}$) which makes error floor problem significant.
- Need for low-complexity low-power decoders.
- Using lower precision in soft-information to improve read latency.

State-of-the-art LDPC Decoder

- Message-passing based on Belief Propagation (BP) or its low-complexity variants
- Variable node update: Sum the incoming extrinsic messages and the channel value, and quantize (typically need 5-6 bits of precision)
- Check node update:
  - Sign Operation: Product of signs of incoming extrinsic messages
  - Magnitude Operation: Find minimum and second minimum of incoming extrinsic messages and apply scaling factor

New Decoding Approach-FAID

- Finite Alphabet Iterative decoding (FAID): Different from Belief Propagation or min-sum-based decoders
- Messages are binary vectors belonging to a small finite alphabet represented as levels 0, $\pm L_1$, $\pm L_2$, etc.
- Size of alphabet is determined by the precision. For 3-bit precision, there are 7 levels.
- Outperforms floating-point BP in error floor with hard-decision (past result) as well as with 2-bit precision soft-information (new result)

FAID: Hard-decision decoding

- Y={-C,+C}, is the set of possible channel values.
- The variable node update Φ_v is a (d_v-1)-dimensional map or look-up table (LUT), where d_v is the column-weight.
- For d_v = 3, it is a 2D LUT. For d_v = 4, it is a 3D LUT.

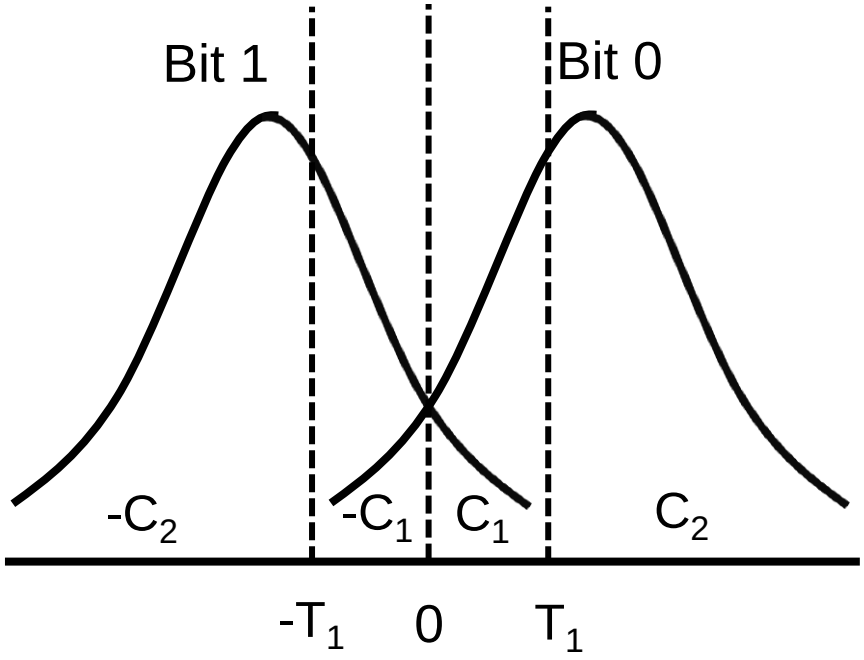
m_1/m_2	$-L_3$	$-L_2$	$-L_1$	0	$+L_1$	$+L_2$	$+L_3$
$-L_3$	$-L_3$	$-L_3$	$-L_3$	$-L_3$	$-L_3$	$-L_3$	$-L_1$
$-L_2$	$-L_3$	$-L_3$	$-L_3$	$-L_3$	$-L_2$	$-L_1$	$+L_1$
$-L_1$	$-L_3$	$-L_3$	$-L_2$	$-L_2$	$-L_1$	$-L_1$	$+L_1$
0	$-L_3$	$-L_3$	$-L_2$	$-L_1$	0	0	$+L_1$
$+L_1$	$-L_3$	$-L_2$	$-L_1$	0	0	$+L_1$	$+L_2$
$+L_2$	$-L_3$	$-L_1$	$-L_1$	0	$+L_1$	$+L_1$	$+L_3$
$+L_3$	$-L_1$	$+L_1$	$+L_1$	$+L_1$	$+L_2$	$+L_3$	$+L_3$

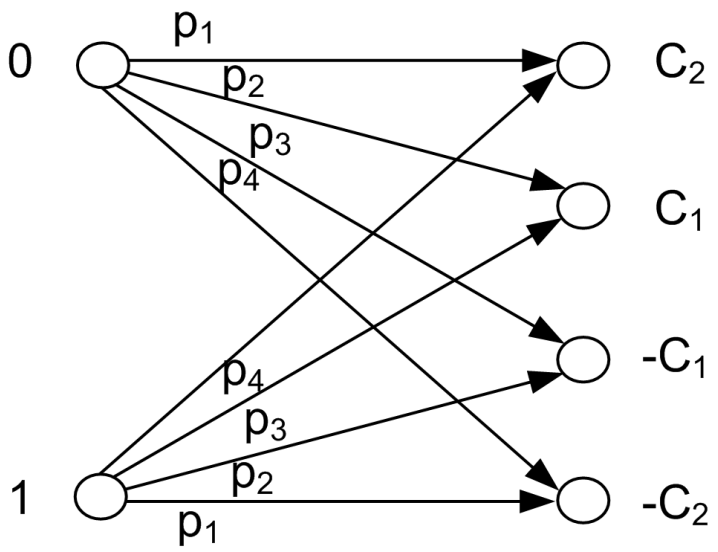
Example of a 2D LUT defining $\Phi_v(+C, m_1, m_2)$ for d_v = 3

FAID: Soft-decision decoding

- The set Y is now Y={$-C_2$,$-C_1$, C_1, C_2}.
- Size of alphabet is still 7 levels (3-bit precision).
- For $d_v = 4$, we now need to design two 3D LUTs, one for $\pm C_1$ and for $\pm C_2$
- They are chosen to optimize for both waterfall and error floor performance.
- We also need to quantize the channel output appropriately.

Channel quantization for 2-bit soft channel information





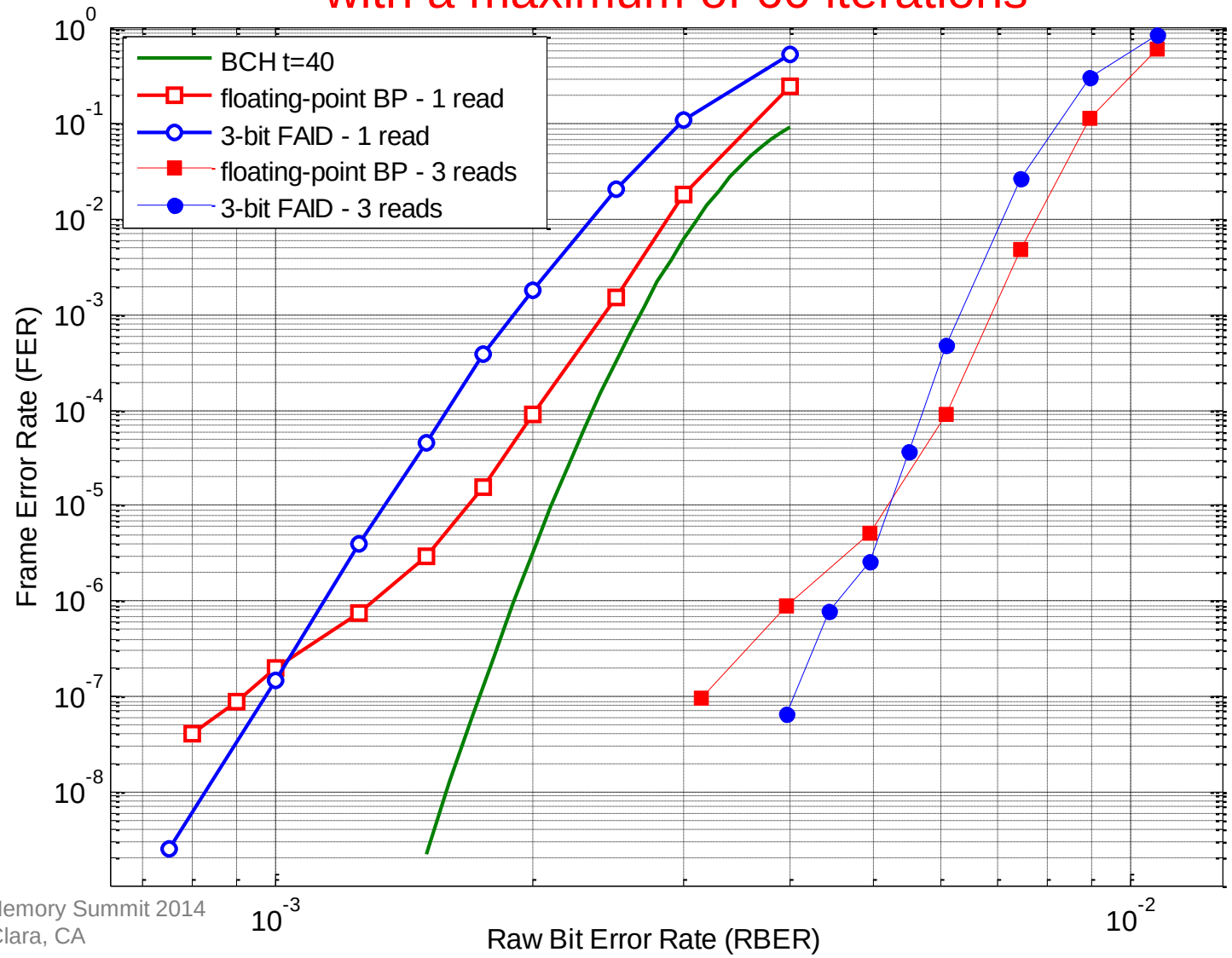
Assuming BPSK + AWGN, the threshold T_1 is chosen for a given SNR so that it maximizes the mutual information of binary-input 4-output channel.

Decoder Complexity

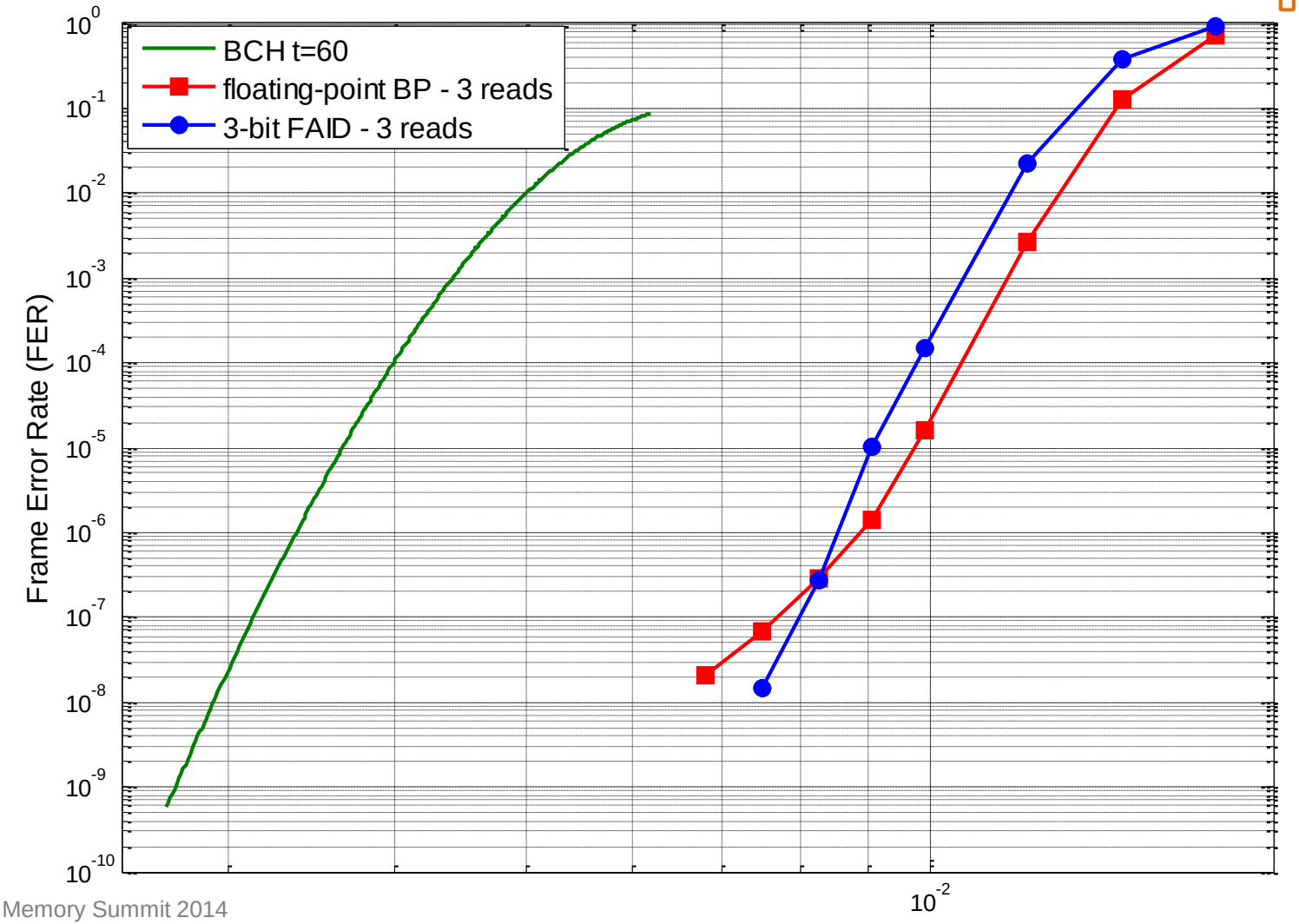
- Low-precision FAIDs lead to significant savings in area and power especially at very high-rates.
- For a d_v=3, R=0.92 code, it was shown in [TCAS'14] that 3-bit FAID could provide more than 50% savings in area. Similar gains or more are expected for d_v=4.
- For 2-bit precision soft-channel, the common entries in the two LUTs of FAID can be exploited for efficient implementations [TCAS'14].

[TCAS'14] F. Cai, X. Zhang, D. Declercq, S. K. Planjery, and B. Vasic, ``Finite alphabet iterative decoders for LDPC codes, optimization, architecture, and analysis," *Trans. Circuits and Systems,* vol. 61, no.5, March 2014.

Performance comparisons: N=1KB, R=0.9375, d_v=4 quasi-cyclic code with a maximum of 60 iterations



Performance comparisons: N=1KB, R=0.91, d_v=4 quasi-cyclic code with a maximum of 60 iterations



Conclusions

Key Benefits

- Ability to perform at very low-precision leading to significant savings in power and area.
- Ability to provide superior error floor performance with minimal loss in waterfall compared to BP and without compromise in decoding latency.
- Ability to perform with low-precision soft-information.

Ongoing Work:

- Extending proof-of-concept of FAIDs to MLC and TLC
- FPGA implementations for further verification.

Acknowledgement

This material presented is based upon work supported by the National Science Foundation under grant no. IIP-1415704.

THANK YOU!