Next Generation ECC Schemes for High-Endurance SSDs

Ravi Motwani

Intel

Non-Volatile Memory Systems Group

Santa Clara, CA, USA

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

All products, computer systems, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

Intel product plans in this presentation do not constitute Intel plan of record product roadmaps. Please contact your Intel representative to obtain Intel's current plan of record product roadmaps.

This document contains information on products in the design phase of development.

Material in this presentation is intended as product positioning and *not* approved end user messaging.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

Results have been simulated and are provided for informational purposes only. Results were derived using simulations run on an architecture simulator or model. Any difference in system hardware or software design or configuration may affect actual performance.

Intel does not control or audit the design or implementation of third party benchmark data or Web sites referenced in this document. Intel encourages all of its customers to visit the referenced Web sites or others where similar performance benchmark data are reported and confirm whether the referenced benchmark data are accurate and reflect performance of systems available for purchase.

Presentation Outline

- History of Modern Codes
- ECC Evolution in Storage
- **Non-Binary LDPC codes**
- **Polar Codes**
  - Reed-Muller Codes (1954)
  - Recursion, Rate of Polarization
  - Decoding polar codes
- Conclusions

History of Modern Codes

- Turbo Codes (1993)
- LDPC Codes (1996)
  - Developed by Gallager in 1960
  - PhD Thesis at MIT
  - http://www.rle.mit.edu/rgallager/documents/ldpc.pdf
- LDPC Code Implementation
  - Accepted for DVB-S2 in 2003
  - Part of Wi-Fi 802.11n (optional) in 2009
  - HDD- Marvel, LSI, BRCM etc put ASIC efforts in 2008
    - Drives with LDPC codes shipped couple years later
    - LDPC codes with 512B information size
    - Marvel, LSI make channels with non-binary LDPC codes

ECC Evolution in Storage

- Hard Disk Drives
  - Reed Solomon Codes
    - – Viterbi detector and burst errors due to defects
  - Binary LDPC Codes
    - – Soft information comes from SOVA
    - – Erasure decoding from media defects
  - Non-Binary LDPC Codes
    - – GF(4), GF(8), GF(16)
- Solid State Drives
  - Algebraic code
    - – BCH codes
  - LDPC codes
    - – Binary LDPC codes
      - – Soft information limited by trigger rate

ECC Evolution in SSDs

- What's next?

- Non-Binary LDPC Codes
  - For HDD, there is inter-symbol-interference (ISI)
  - ISI makes non-Binary LDPC codes suitable for HDD

- Polar Codes
  - Recent results show they have potential

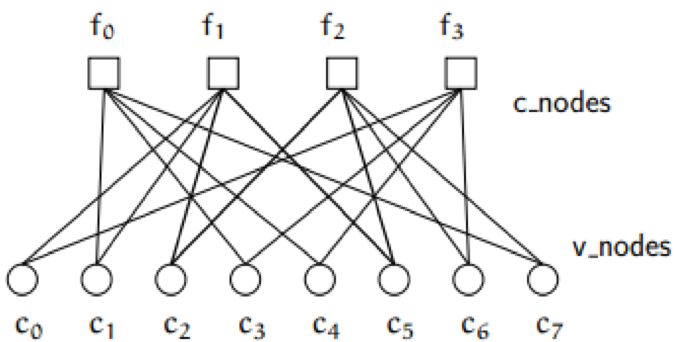
Non-Binary LDPC codes

- Instead of working on bits, non-binary LDPC codes work on groups of bits (called symbols)
- Symbols can be a set of $1,2,\ldots q$ bits
- Galois fields- $GF(2^2), GF(2^4),$ …… $GF(2^q)$

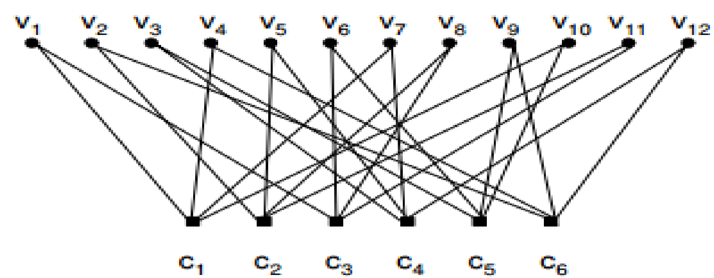
Parity check matrix

- H-matrix of a binary vs non-binary LDPC code over GF(8)

$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$



$$H = \begin{bmatrix} 5 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 3 & 0 & 0 & 3 & 0 & 0 & 5 & 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 0 & 0 & 5 & 0 & 2 & 0 & 0 & 7 & 0 \\ 0 & 0 & 5 & 0 & 6 & 0 & 3 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 2 & 0 & 0 & 3 & 0 & 0 & 1 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 4 \end{bmatrix}$$



All operations are over $GF(2^q)$

Non-Binary LDPC code and its binary representation

- Any non-binary LDPC code can be represented by its binary equivalent

$$H = \begin{bmatrix} 5 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 3 & 0 & 0 & 3 & 0 & 0 & 5 & 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 0 & 0 & 5 & 0 & 2 & 0 & 0 & 7 & 0 \\ 0 & 0 & 5 & 0 & 6 & 0 & 3 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 2 & 0 & 0 & 3 & 0 & 0 & 1 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 4 \end{bmatrix}$$

- Replace all the $GF(2^3)$ entries by their 3 x 3 binary equivalents

Why the difference then?

- Encoding/Decoding done in $GF(2^q)$
- Message passing works on symbol basis
- All properties of the code are in that space
- Girth, distance properties
  - Typically large girths with small column weights
- Binary representation helps with code construction

Why non-binary LDPC should perform better for SDD

- Hard disk drives
  - Have ISI
- Even for AWGN channels, literature on non-binary LDPC codes shows improved performance

Decoding non-binary LDPC codes

- Binary LDPC codes
  - Min-Sum Decoder, 2-D Min-Sum Decoder
- Non-Binary LDPC codes
  - Extended Min-Sum (EMS) decoder
- Message Passing Algorithms
  - Probability domain
    - Check node update in Fourier domain- FFT
  - Log domain

Decoding non-binary LDPC codes

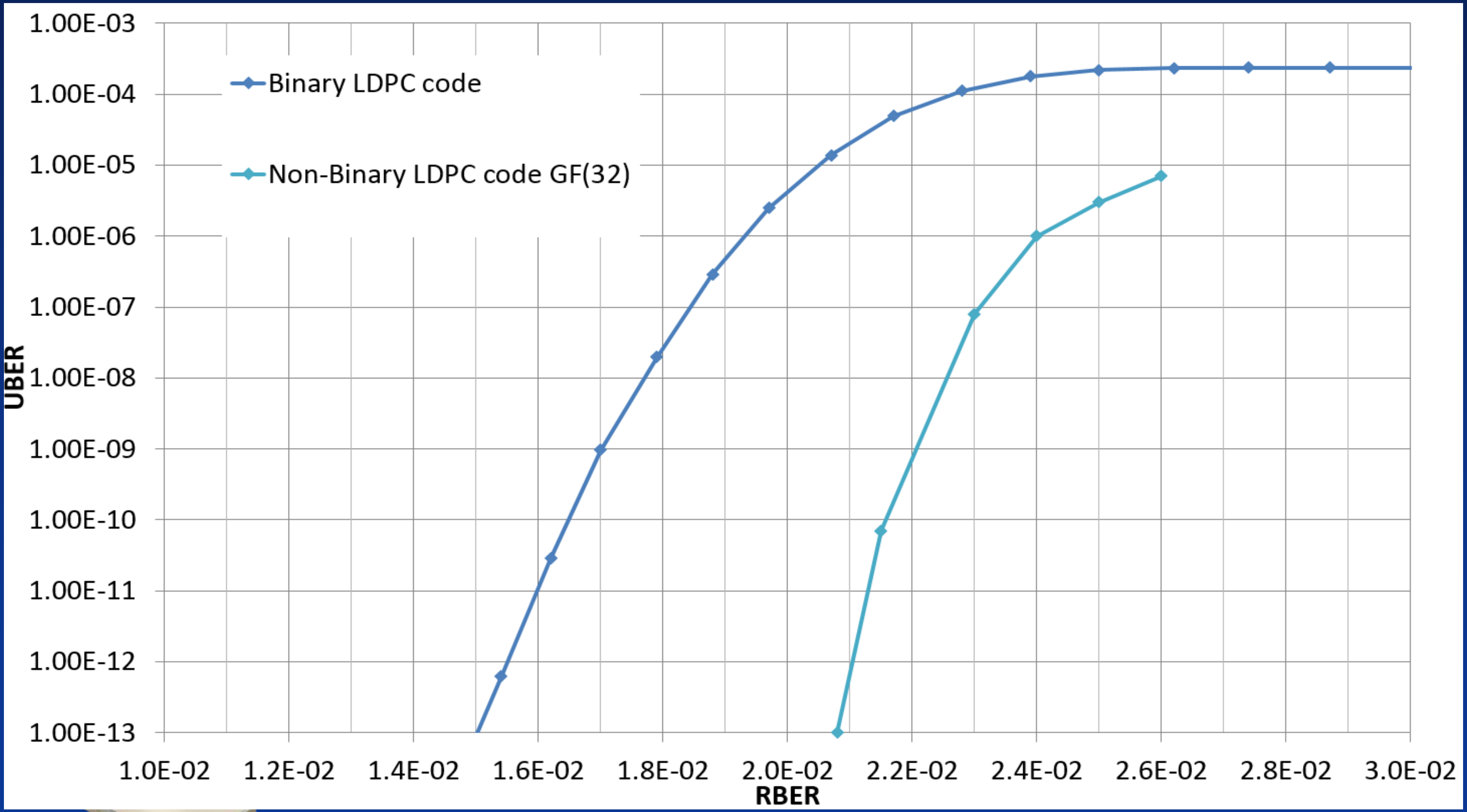
- Log-density-ratios (LDR)
- $LDR(s) = \log \frac{p(r|s)}{p(r|0)}, s = 0,1,..2^q - 1$
- From r, compute the $LDR(s)$
- Message passing consists of updating the LDRs at the check and symbol nodes
- Introduce permutation nodes

Decoding Non-Binary LDPC codes

- Symbol flipping algorithm
  - Bit flipping decoding for binary LDPC codes
- Min-max decoding
  - Simplified decoding
- Trellis EMS algorithm
  - Ideal for high throughput, high rate applications
  - Memory requirements are huge

Simulation Results

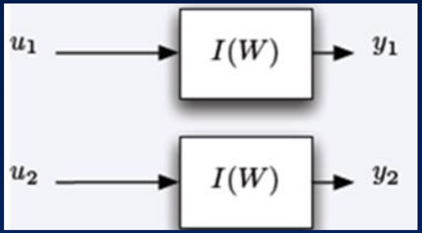
- 1KB LDPC codewords, soft decision decoding, simulation results at Intel- 1.53x RBER gain



Polar Codes- History

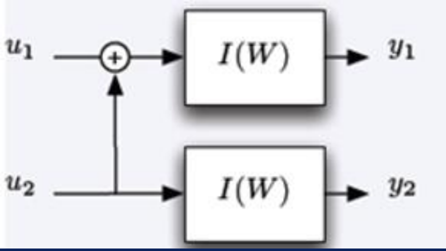
- Erdal Arikan- 2008
- Binary discrete memory-less channels (B-DMC)
- Capacity achieving codes with low encoding and decoding complexity- $O(N\ logN)$
- Minimum codeword size for channels to polarize
  - 2K bits
- Successive cancellation decoding algorithm
- List Decoding with CRC- Tal & Vardy

Channel Polarization



- $I(u_1, u_2; y_1, y_2) = I(u_1; y_1, y_2) + I(u_2; y_1, y_2 | u_1)$
  $= I(u_1; y_1) + I(u_2; y_2)$
  $= I(W) + I(W) = 2I(W)$
- Synthesize two channels from two independent copies of DMC channels *W*
- The two channels have same symmetric capacity

Channel Polarization



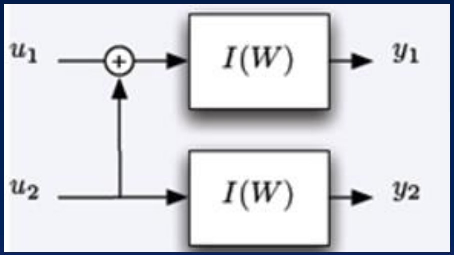
- $I(u_1, u_2; y_1, y_2) = I(u_1; y_1, y_2) + I(u_2; y_1, y_2 | u_1)$

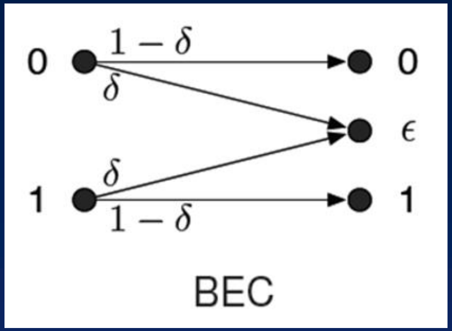
$$= I(W') + I(W'') = 2I(W)$$

$$I(W') \le I(W) \le I(W'')$$

- Created two channels
- One channel can have higher capacity than the other
- Total capacity of the two channels is unchanged

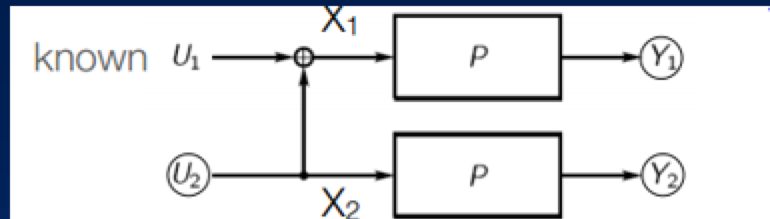
Channel Polarization- BEC



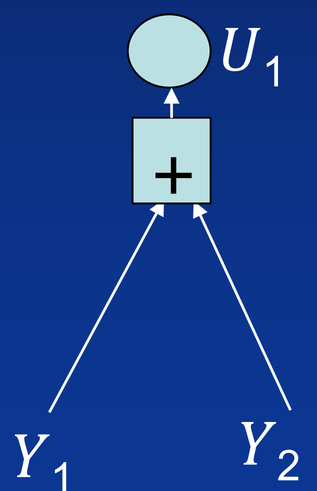


- u_1 is erased if either y_1 or y_2 is erased
- u_2 is erased if both y_1 or y_2 are erased
- Probability of u_1 erased is $2\delta(1-\delta)+\delta^2$
- Probability of u_2 erased is δ^2
- δ=0.4, $I(W)$ =0.6
- P(u_1 erased)=0.64, $I(W')$= 0.36 $< I(W)$
- P(u_2 erased)=0.16, $I(W'')$=0.84 $> I(W)$

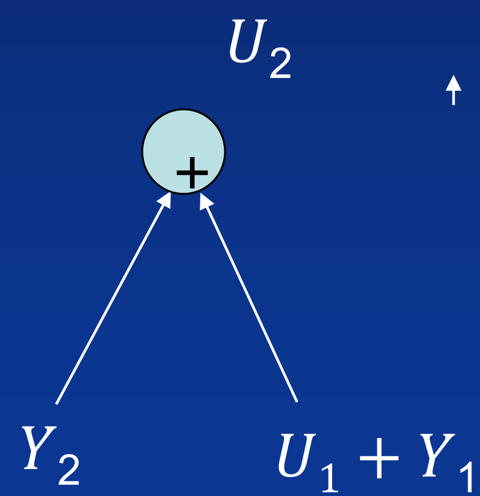
Why channels polarize?



- $Observe \;\; Y_1, Y_2$
- $U_1 = X_1 + X_2$
- $U_2 = X_2; U_2 = X_1 + U_1$



Parity check node



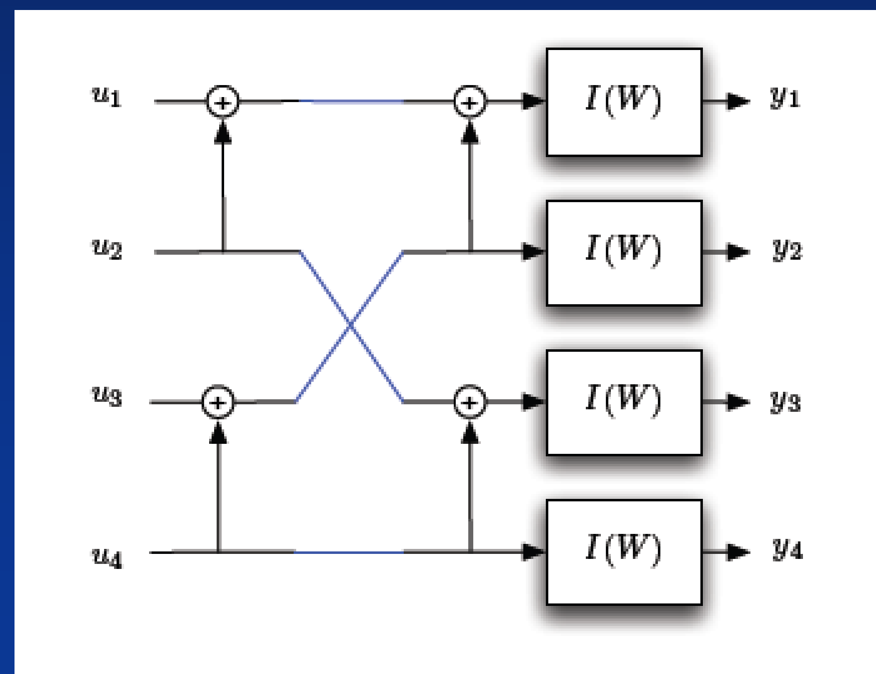
Variable node
Repitition code

Polar Codes

- $m = 2$
- Recursive code construction
- Kronecker Product to get $N = 4$

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

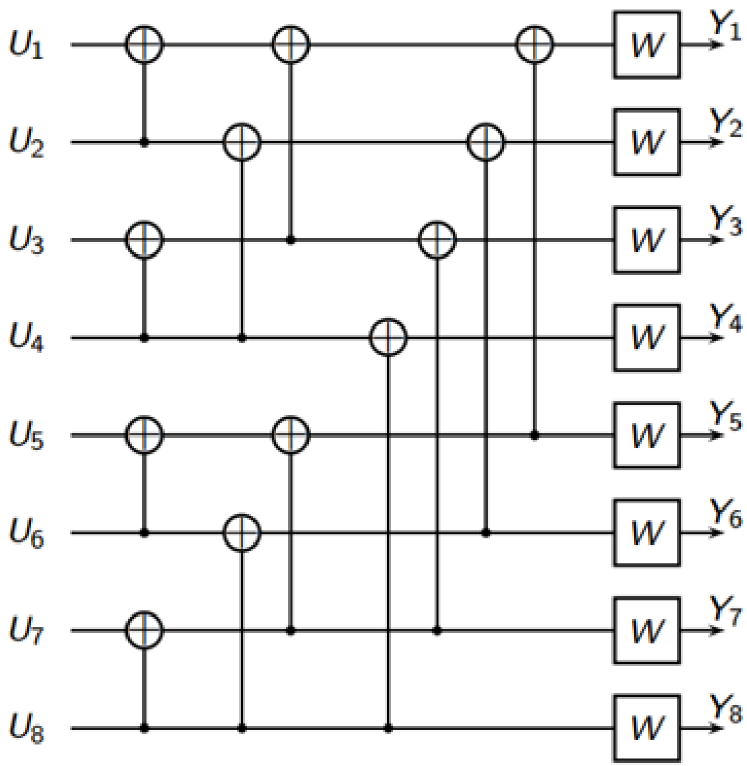
$$G_2^{\otimes 2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$



Codes from Kronecker Products of G_2

length $N = 2^m$, $m \in \mathbb{N}$

generator matrix: rows of $G_2^{\otimes m}$



$$G^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Reed-Muller Codes

length $N = 2^m$, $m \in \mathbb{N}$

generator matrix: rows of $G_2^{\otimes m}$

How to choose the rows?

$$G_2^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

choose rows of largest weight

$$\mathbf{u} = (0, 0, 0, u_4, 0, u_6, u_7, u_8)$$

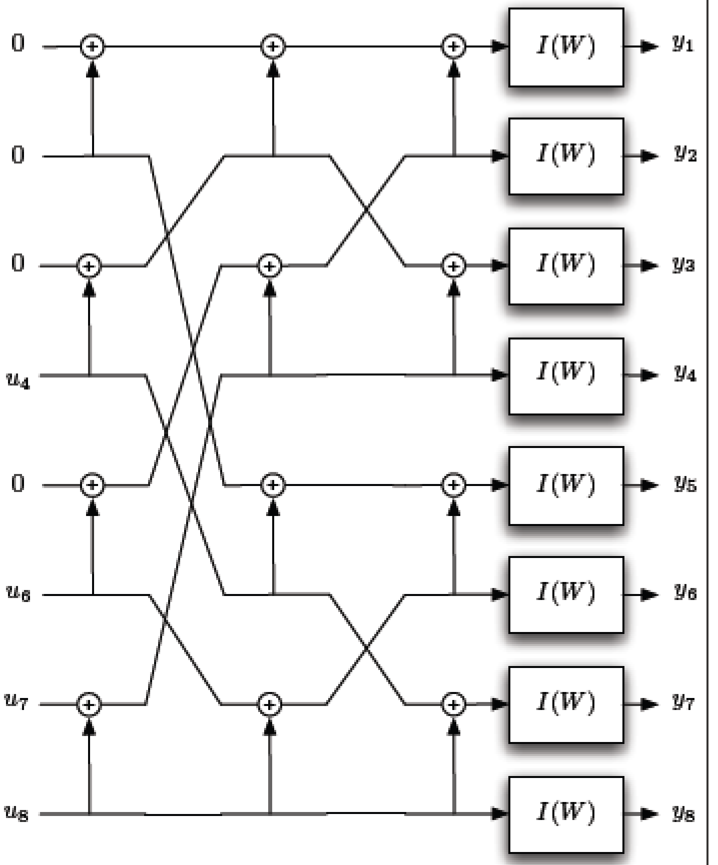
Polar Codes

length $N = 2^m$, $m \in \mathbb{N}$

generator matrix: rows of $G_2^{\otimes m}$

$$G_2^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\bar{x} = (0, 0, 0, u_4, 0, u_6, u_7, u_8)\ G_2^{\otimes 3}$$



Frozen set

- Freeze the bits on the bad channel- Frozen set
- Useless Channels, asymptotically $W_N^{(i)}\left(y_1{}^N, u_1{}^{i-1} \middle| u_i\right) = 0.5, u_i = 0,1$
- These indices i are the ones which are channels with capacity 0

Polar Codes

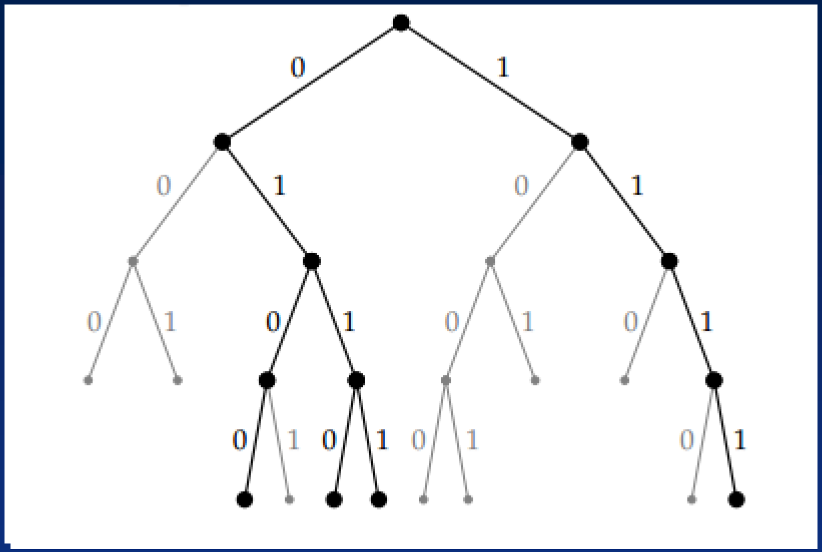
- Choice of frozen set
  - RM- Choose the rows with maximum Hamming weight
  - Bhattacharya parameter
- Only for code lengths which are powers of 2
  - Shortening
  - Other base matrices or combinations
- Decoding
  - Successive Cancellation Decoding
  - List Decoding + CRC
- Non-systematic codes
  - Can we do systematic constructs?

Shortening Polar Codes

- Default length of polar codes is 2^q, for some integer q
- Is shortening possible*?
- Yes, since the generator matrix is a lower triangular matrix
- Hard decision decoding shows RBER advantage and quite some endurance benefit *

- * Yue Li et al, “The performance of Polar Codes for Multi-level Flash Memories,” NVM Workshop 2014

List Decoding of Polar Codes with CRC



- List size has to be at least 32 or more
- Decoder memory impact since we need to store n codewords in the list
- Not as amenable to decoding as LDPC codes
- Multiple rate constructs difficult

Conclusions

- Non-Binary LDPC codes are an appropriate future generation choice
- Polar codes competing with non-binary LDPC codes?
  - Not beating non-binary LDPC codes on RBER
  - Polar codes not as amenable to decoding as non-binary LDPC codes
  - Variable rate constructs not as easy as LDPC
  - List size is large which has SRAM cost downsides