

Error-Correcting Codes in 5G/NVM Applications

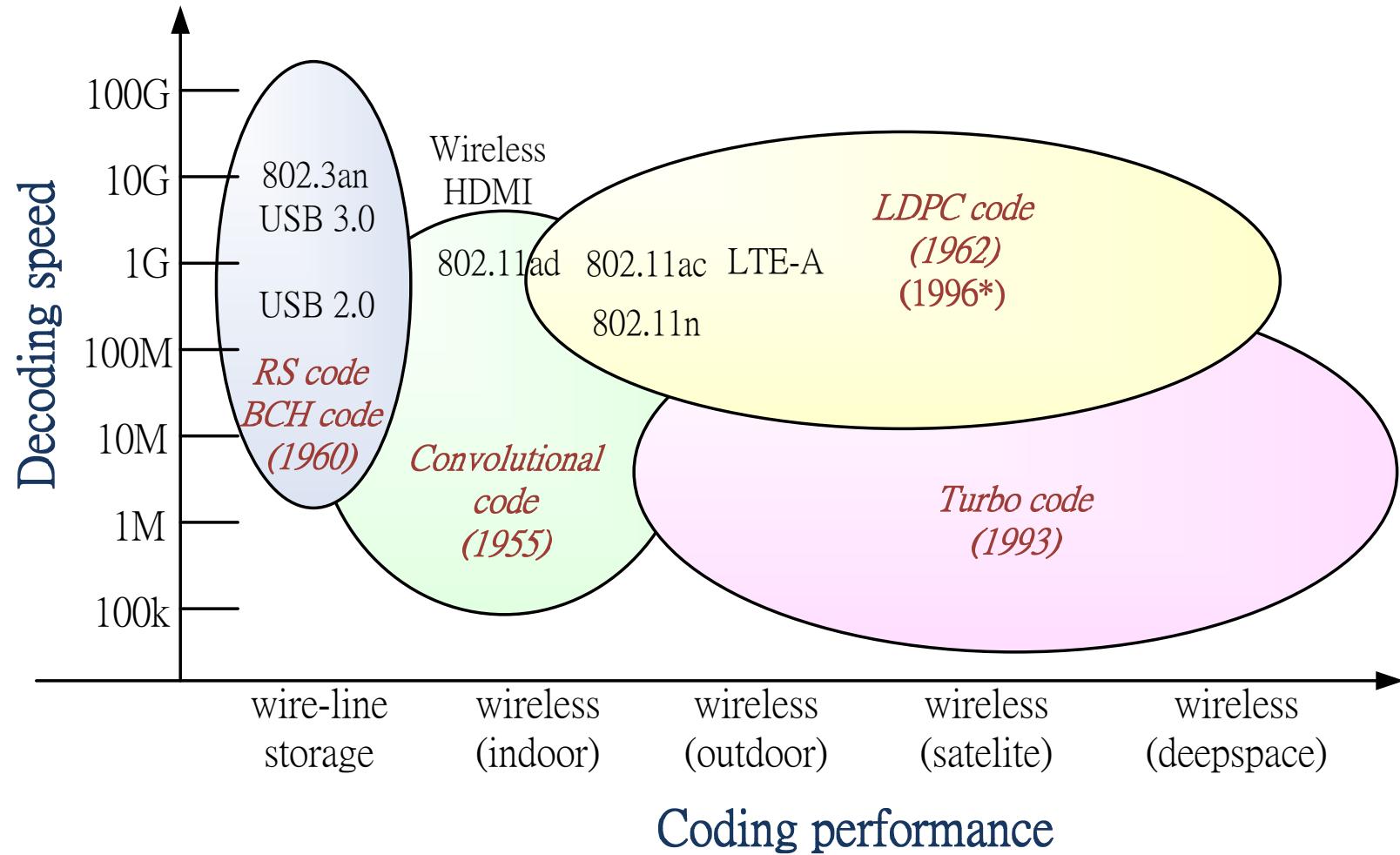
Hsie-Chia Chang

hcchang@mail.nctu.edu.tw

National Chiao Tung University, Taiwan



Error-Correcting Code (ECC) Constellations



Error-Correcting Code (ECC) Constellations

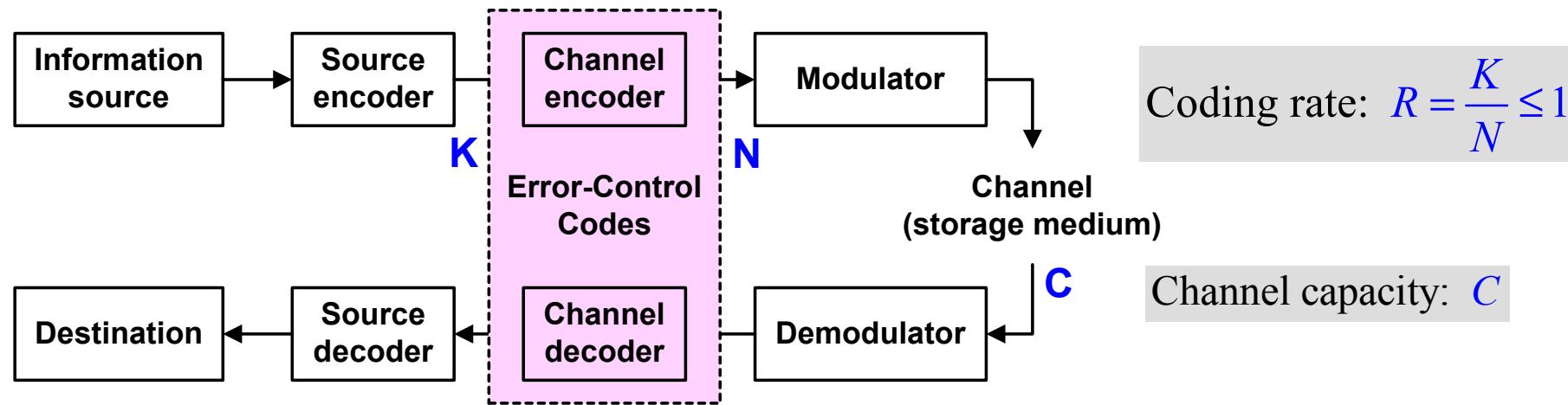
The growing needs of efficient data transmission and storage drive the information delivery technology to a new frontier recently. Either in the new radio (NR) links of 5G communications, or in the emerging non-volatile memories (NVMs) with continuously increasing capacity, error correcting codes (ECCs) are essential and crucial in maintaining the data correctness.

In this tutorial, we will cover major ECC design and architecture, including over Gb/s LDPC-BC, energy-efficient LDPC-CC, and Polar/Turbo decoders that fulfill different requirements in various 5G scenarios. For NVM applications, we will introduce 1-error and 2-error correcting scheme with parallel architecture for NOR flash, as well as BCH and LDPC coding scheme for NAND/3D-NAND, to address low-latency and high-throughput solutions.

Outline

- **ECC Primer**
 - FEC = ECC + Randomization + Interleaving
 - **ECC in 5G Communications Systems**
 - Turbo Codes vs. LDPC Codes
 - Polar Codes in short block length
 - **ECC in Non-volatile Memory (NVM) Systems**
 - BCH Codes vs. LDPC Codes
 - Soft Information in NAND Flash Memory
 - **Summary**
 - **Reference**
-

ECC Primer: Channel Encoder/Decoder



- Recently, there has been an increasing demand for efficient and reliable digital data transmission and storage systems
- A major concern of the designer is the control of errors so that reliable reproductions of data can be obtained
- In 1948, Shannon claims that errors induced by a noisy channel can be reduced to any desired level by proper encoding of the information

ECC Primer: Shannon Capacity

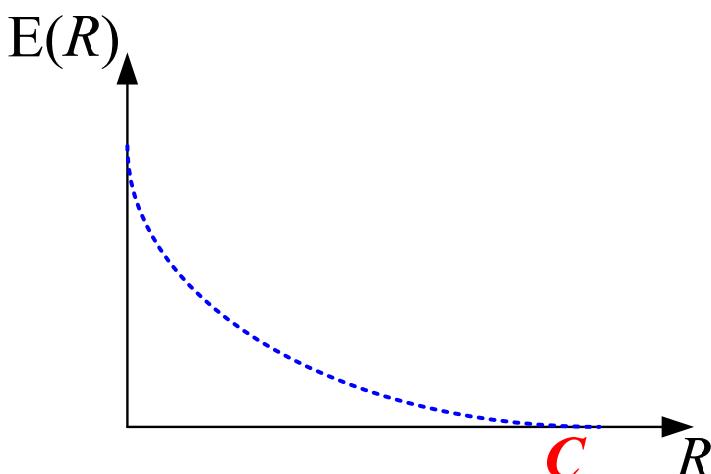
- Every channel has a capacity C
- For every rate $R < C$, there exists a code that can be decoded with error probability $\rightarrow 0$ as $N \rightarrow \infty$

- Block codes with length N ,

$$P(E) \leq 2^{-NE_b(R)}$$

- Convolutional codes

$$P(E) \leq 2^{-(M+1)NE_c(R)}$$



- $E_b(R)$ and $E_c(R)$ are positive functions of R and are completely determined by the channel characteristics

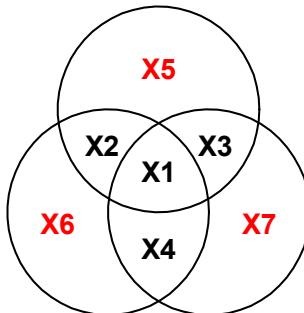
ECC Primer: Block Codes

- A (N, K) block code contains K message symbols independently into N coded symbols, which is called a codeword

- Code rate = K/N



- For example, the $(7, 4)$ Hamming code

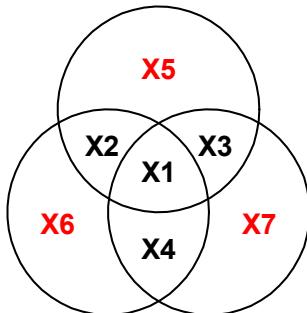


ECC Primer: Block Codes

- A (N, K) block code contains K message symbols independently into N coded symbols, which is called a codeword
- Code rate = K/N



- For example, the $(7, 4)$ Hamming code



$$\begin{aligned}x_5 &= x_1 + x_2 + x_3 \\x_6 &= x_1 + x_2 + x_4 \\x_7 &= x_1 + x_3 + x_4\end{aligned}$$

ECC Primer: Block Codes

- A (N, K) block code contains K message symbols independently into N coded symbols, which is called a codeword
- Code rate = K/N



- For example, the $(7, 4)$ Hamming code

A Venn diagram consisting of four overlapping circles. The regions are labeled: x_5 (top), x_6 (bottom-left), x_7 (bottom-right), and x_1, x_2, x_3, x_4 (the intersections of the circles).

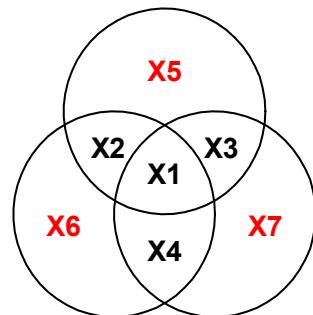
$$\mathbf{x} = \mathbf{u} \cdot \mathbf{G} = [x_1 \ x_2 \ x_3 \ x_4] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

ECC Primer: Block Codes

- A (N, K) block code contains K message symbols independently into N coded symbols, which is called a codeword
- Code rate = K/N



- For example, the $(7, 4)$ Hamming code

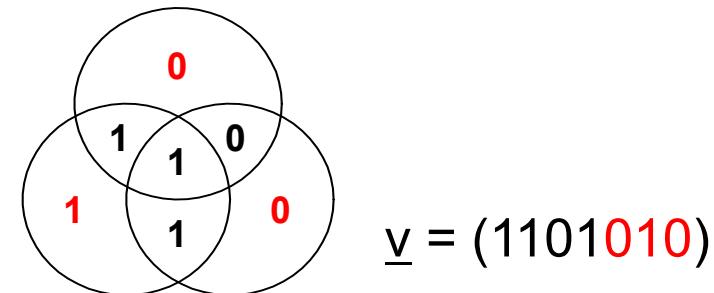


$$x_5 = x_1 + x_2 + x_3$$

$$x_6 = x_1 + x_2 + x_4$$

$$x_7 = x_1 + x_3 + x_4$$

$$\vec{v} = \vec{u} \cdot G = [x_1 \ x_2 \ x_3 \ x_4] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

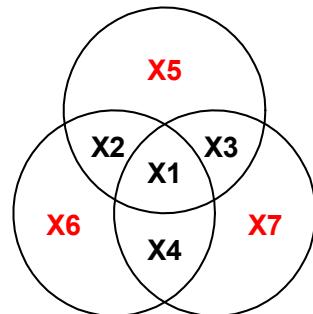


ECC Primer: Block Codes

- A (N, K) block code contains K message symbols independently into N coded symbols, which is called a codeword
- Code rate = K/N



- For example, the $(7, 4)$ Hamming code

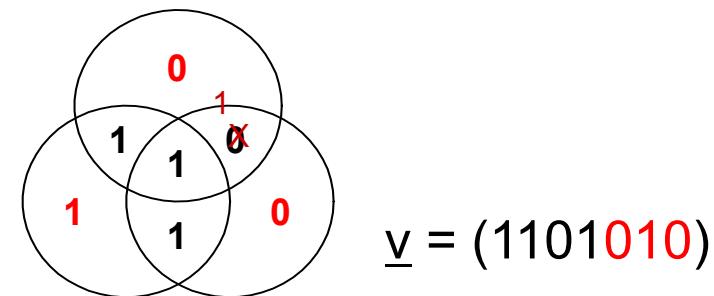


$$x_5 = x_1 + x_2 + x_3$$

$$x_6 = x_1 + x_2 + x_4$$

$$x_7 = x_1 + x_3 + x_4$$

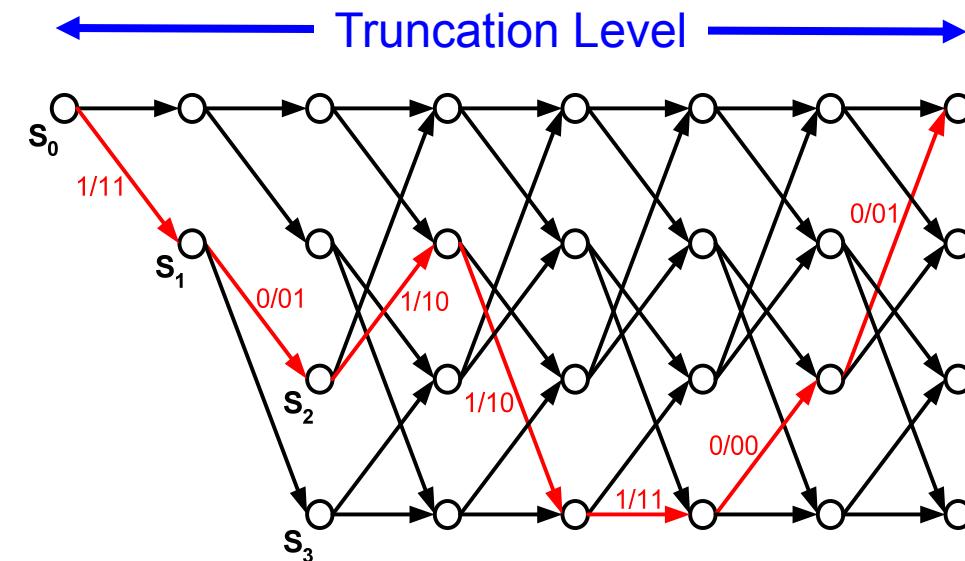
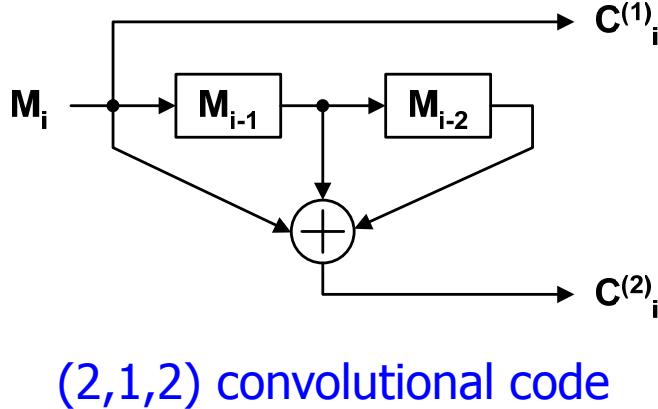
$$\vec{v} = \vec{u} \cdot G = [x_1 \ x_2 \ x_3 \ x_4] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$



While receiving $r = (11\textcolor{red}{1}1010)$,
the error can be easily
corrected.

ECC Primer: Convolutional Codes

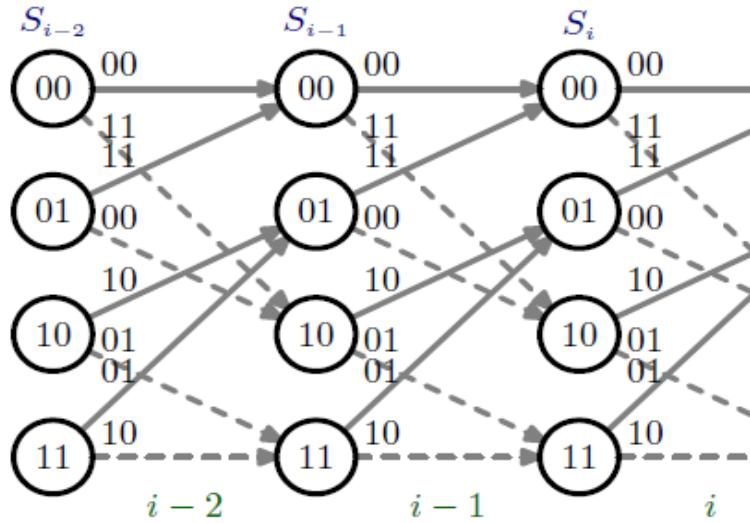
- A (N, K, M) convolutional code depends not only on the current K bits, but also on M previous message bits



- Code rate = K/N
- Trellis diagram is used to find the maximum likelihood path

ECC Primer: Convolutional Codes

State diagram / Trellis diagram



$$\begin{aligned} u_i \in \{0, 1\} &\rightarrow (c_i^{(0)}, c_i^{(1)}, \dots, c_i^{(n-1)}), c_i^{(j)} \in \{0, 1\} \\ &\rightarrow (y_i^{(0)}, y_i^{(1)}, \dots, y_i^{(n-1)}), y_i^{(j)} = (-1)^{c_i^{(j)}}, y_i^{(j)} \in \{+1, -1\} \\ &\rightarrow (r_i^{(0)}, r_i^{(1)}, \dots, r_i^{(n-1)}), r_i^{(j)} = y_i^{(j)} + e_i^{(j)}, e_i^{(j)} \sim (0, \sigma^2), r_i^{(j)} \in \mathbb{R} \\ y &= (y_0, y_1, \dots, y_{N-1}) \\ e &= (e_0, e_1, \dots, e_{N-1}) \\ r &= y + e = (r_0, r_1, \dots, r_{N-1}) \end{aligned}$$

- Maximum Likelihood (ML) Algorithm

- Probability: $\Pr(r | y) = \prod_{i=0}^{N-1} \Pr(r_i | y_i)$
- Distance: $\gamma(r; y) = \sum_{i=0}^{N-1} \|r_i - y_i\|$
- Objective: $\max_y \{\Pr(r | y)\} \Leftrightarrow \min_y \{\gamma(r; y)\}$

FEC = ECC + Randomization + Interleaving

- Randomization
 - Randomizer provides for even distribution of the data on the channel to allow effective QAM demodulation
 - A PN (pseudorandom noise) sequence is usually XOR-ed to the data for avoiding continuous "1"s or "0"s
 - **Randomizer is also called Scrambler**
- Interleaving
 - Interleaving mechanism evenly disperses the symbols to protecting against a burst of symbol errors
 - **Block interleaving**
 - **Convolutional interleaving**
- FEC (Forward error correction) mechanism is included
 - error-control codes, randomization, and interleaving

Outline

□ *Introduction*

- *FEC = ECC + Randomization + Interleaving*

□ **ECC in 5G Communications Systems**

- Turbo Codes vs. LDPC Codes
- Polar Codes in short block length

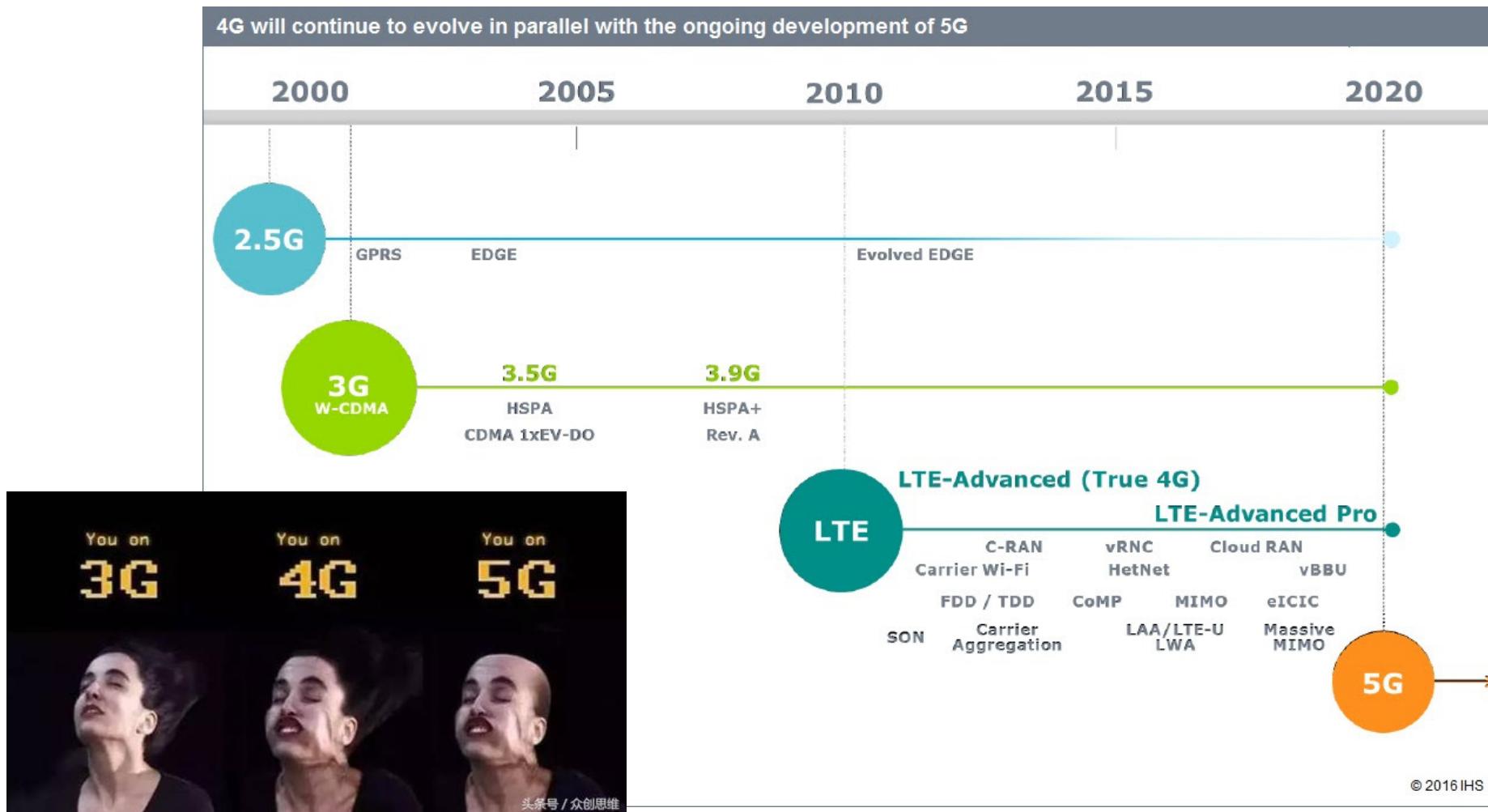
□ *ECC in Non-volatile Memory (NVM) Systems*

- *BCH Codes vs. LDPC Codes*
- *Soft Information in NAND Flash Memory*

□ *Summary*

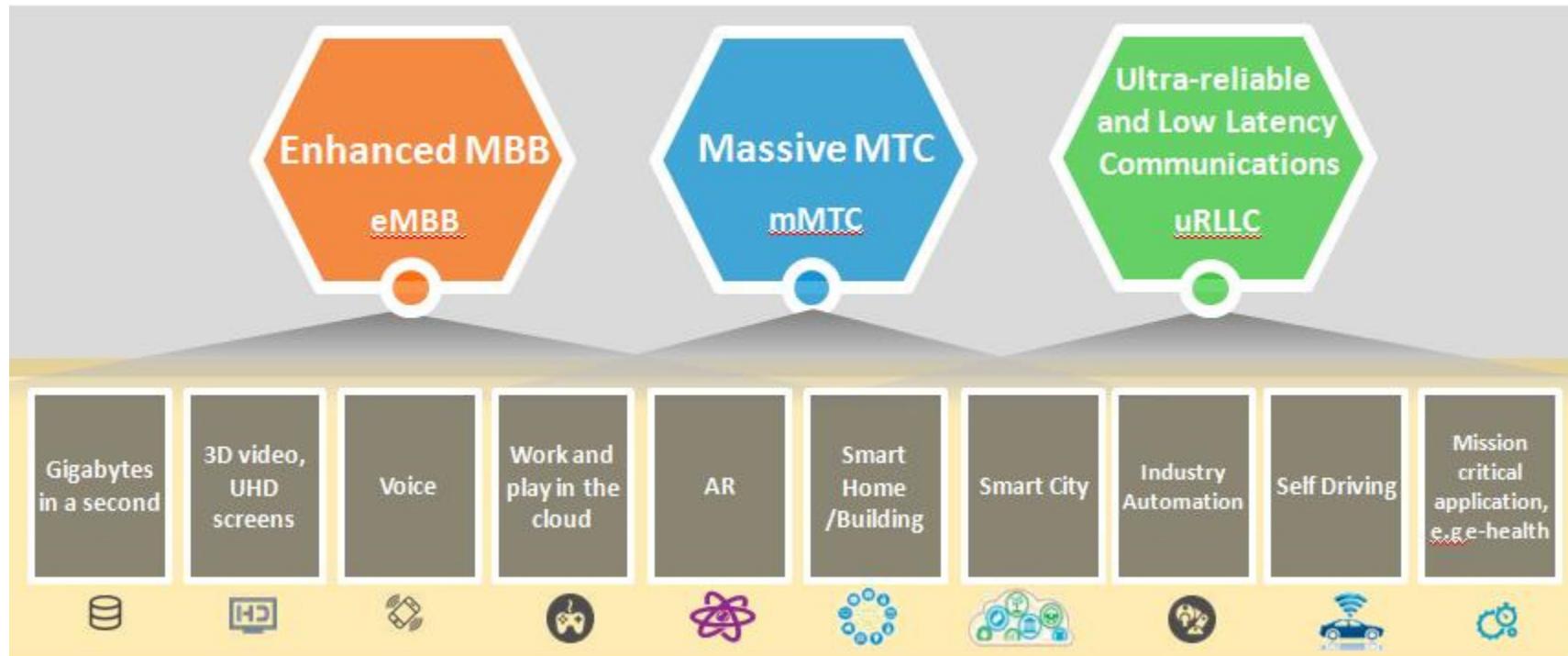
□ *Reference*

What is 5G?



New Radio (NR)

- Wireless standard will become the foundation for the next generation of mobile networks

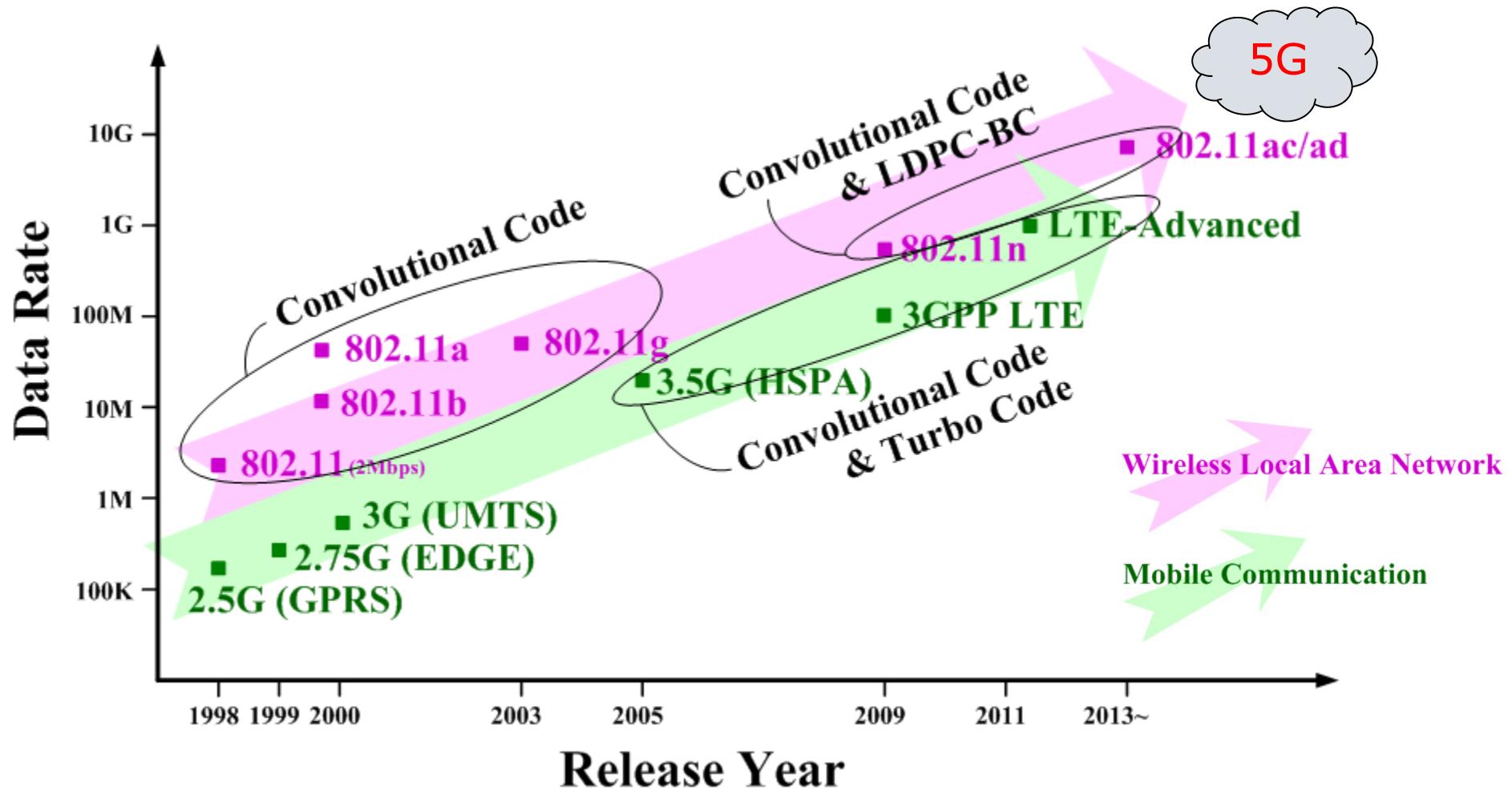


■ **Whereas 3G and 4G connected people, 5G will connect everything**

ECC in wireless communication systems

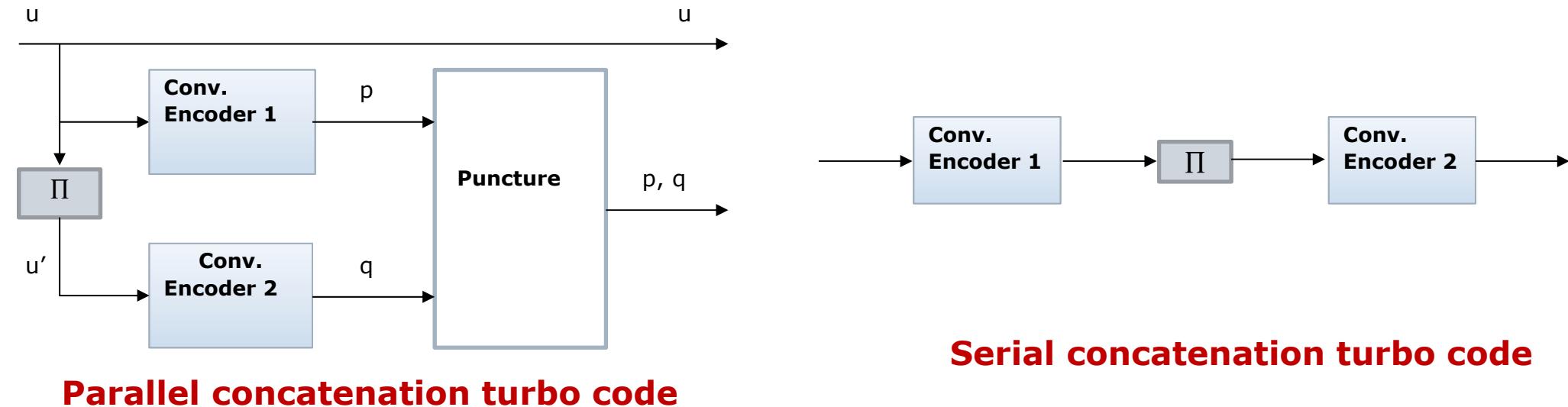
- eMMB – enhanced Mobile BroadBand
 - > 6GHz, small cell
 - **Throughput** (20Gbps/10Gbps)
 - uRLLC – ultra Reliable and Low Latency Communication
 - Self-driving car
 - **Latency** (< 1ms)
 - **Reliability** (< 10^{-5})
 - mMTC – massive Machine Type Communication
 - Smart building/home/city
 - **Cost & Power efficiency**
-

ECC in wireless communication systems



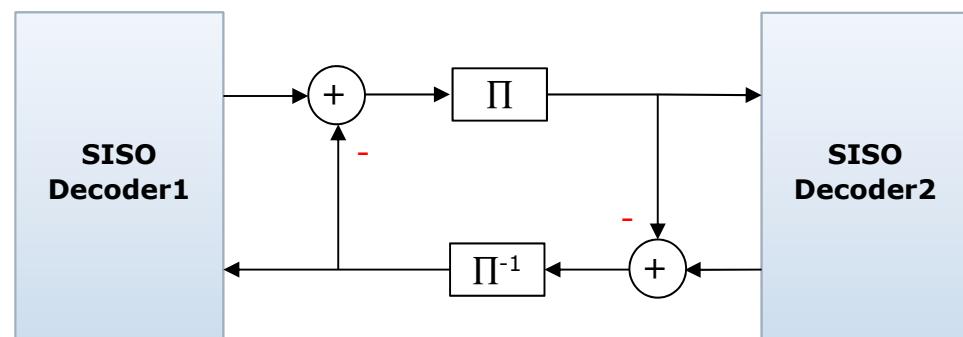
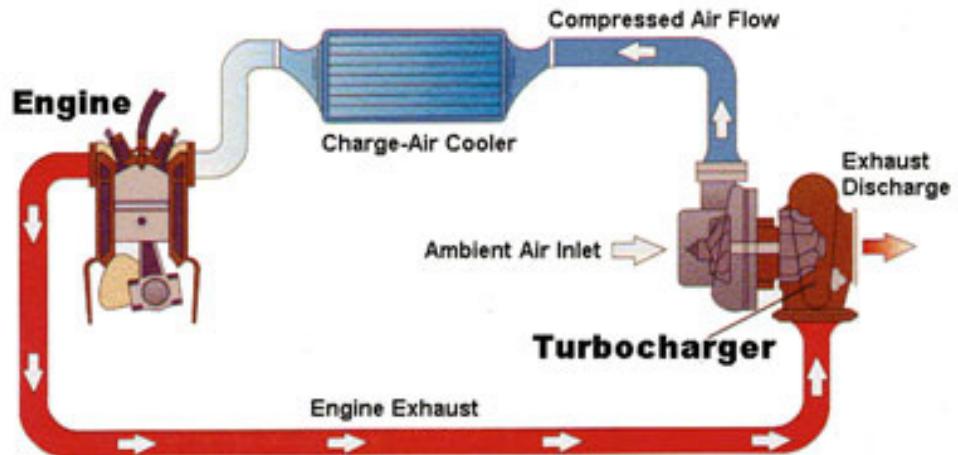
Turbo Codes

- Berrou etc., proposed the first turbo code in 1993
 - Two parallel concatenated convolutional codes (PCCC)
 - A break through in coding community of the revolutionary **iterative** “turbo principle” (ex. turbo decoding, turbo equalization)
 - First near-Shannon-limit code
- The turbo principle applies to both parallel and serial concatenation



The Turbo Principle

- Turbo code got the name because the decoders use the **feedback information**
- The **extrinsic information** of the coded bits are updated iteratively
 - Soft-in-soft-output (SISO) decoding (ex. **maximum a posterior** decoding)
- The interleaver design impacts the performance
 - Random interleaving has better error correction capability
 - Limitations in implementation



Low Density Parity Check Block Codes (LDPC-BC)

- Invented by Roger Gallager in his doctoral dissertation in 1960 [1]
- Generalized by Tanner with a graphical representation in 1981 [2]
- Re-discover by Mackay in the mid. 1990 [3-6]
- Capacity-approaching with large block length
- LDPC is a class **linear block code**
 - With a large and **sparse** parity check matrix, which can be constructed either randomly (by computer search) or by algebraic methods
 - **Tanner graph** (a bipartite graph) representation of the parity check matrix

$$H = \begin{bmatrix} V_0 & V_1 & V_2 & V_3 & V_4 & V_5 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{matrix}$$

A **check node Ci** is associated to a parity check equation

A **variable node Vj** is associated to a coded bit

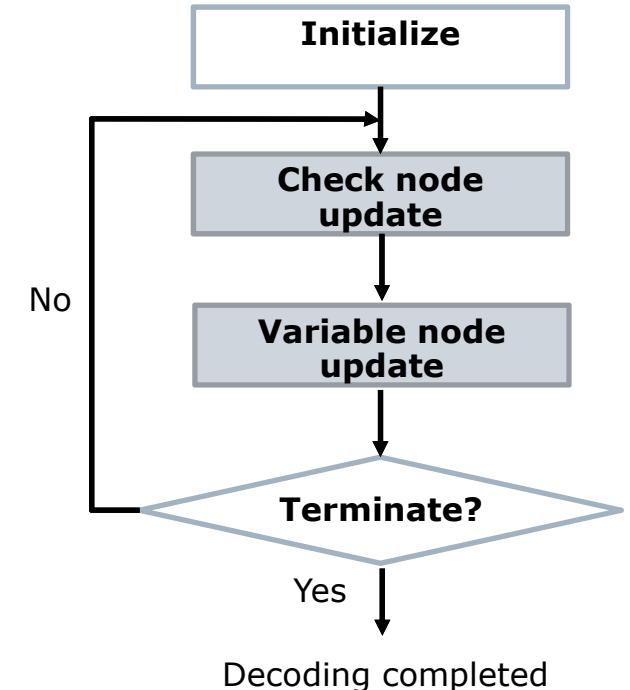
Message Passing Decoding for LDPC

❑ Belief-propagation decoding

- the reliability of each decoded bit is updated and increased after several runs of decoding iteration.
- Each iteration comprises “Check node” and “Variable node” update
- **Termination condition** indicates all parity checks are fulfilled or iteration number exceeds a predefined number

❑ Inherent parallelism facilitates high throughput design

- Fully parallel: all check nodes and variable nodes are updated concurrently
- Partial parallel: with properly schedule may improve the decoding convergence



Turbo Codes vs. LDPC-BC Codes

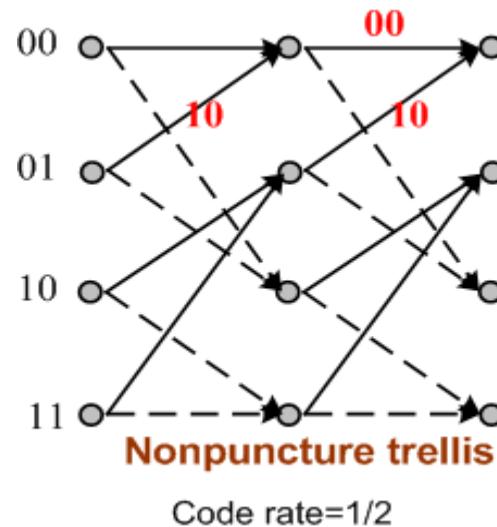
	Turbo Codes	LDPC Codes
Encoding complexity	Lower	Higher
Iterative Decoding algorithm	MAP/SOVA algorithm	Belief propagation
Decoding parallelism	low	high
Early termination by exact valid codeword	No	Yes (by parity check results)
Implementation challenge	Data dependency	Routing complexity

Dual Trellis Turbo Decoder Design (1/3)

□ Turbo Codes

- Reciprocal dual trellis has shorter trellis stage and simpler structure

□ Example :

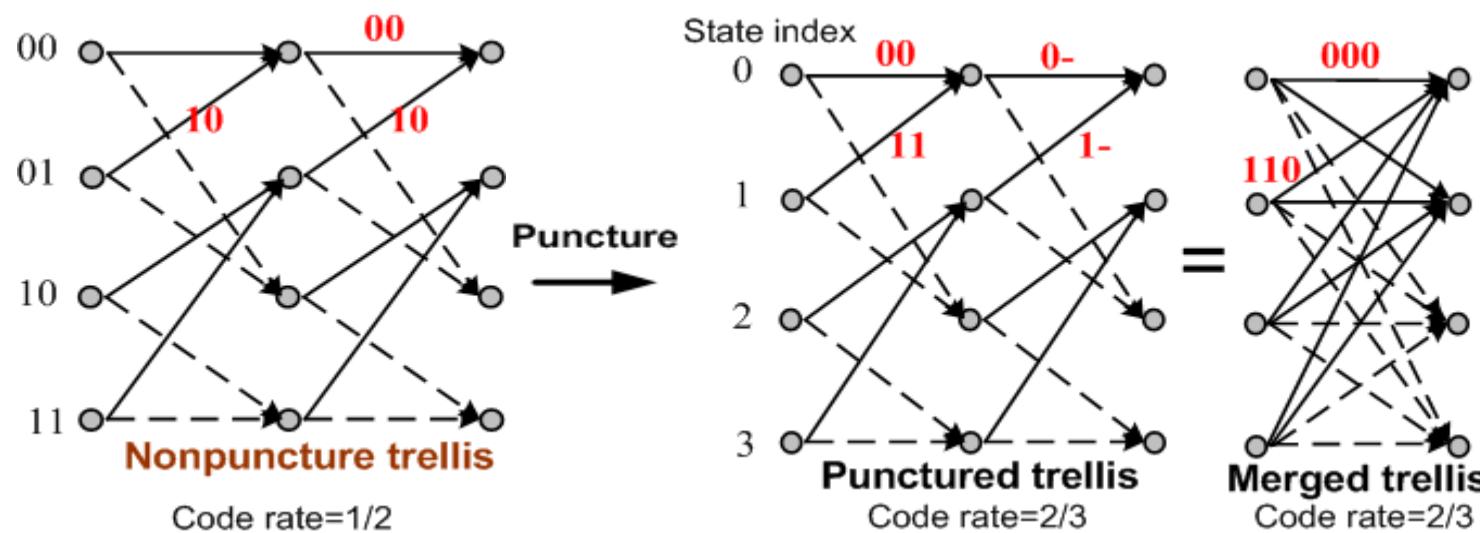


Dual Trellis Turbo Decoder Design (1/3)

□ Turbo Codes

- Reciprocal dual trellis has shorter trellis stage and simpler structure

□ Example :

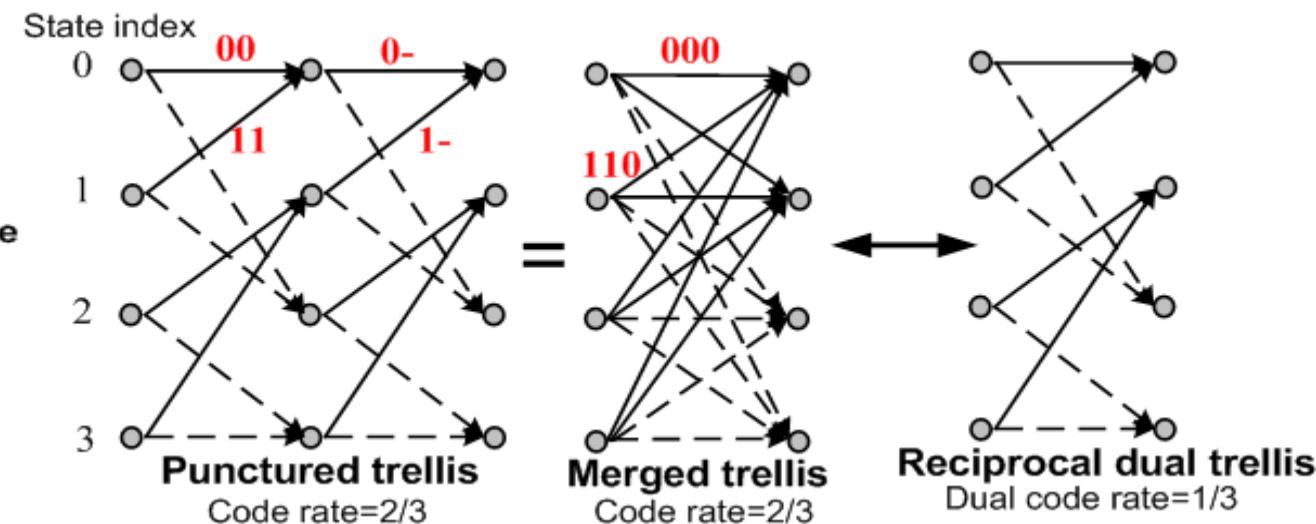
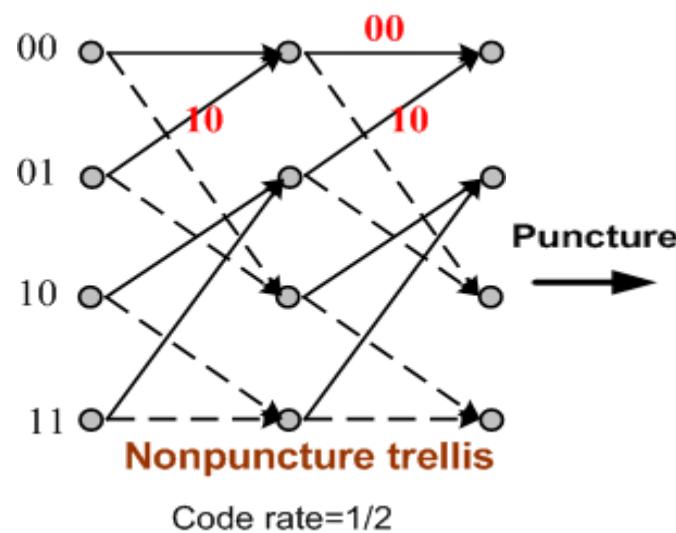


Dual Trellis Turbo Decoder Design (1/3)

□ Turbo Codes

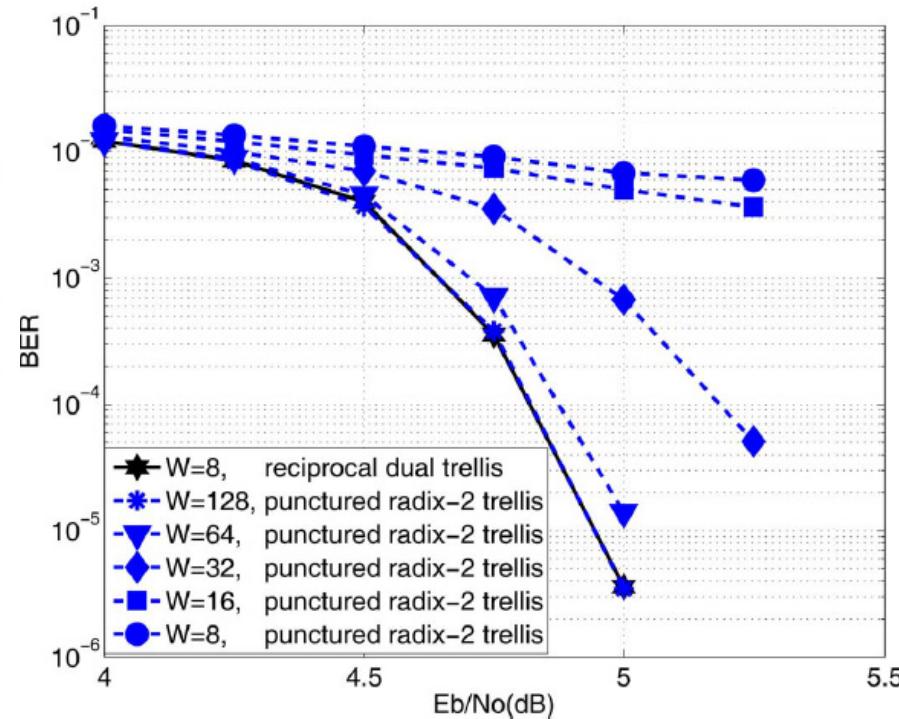
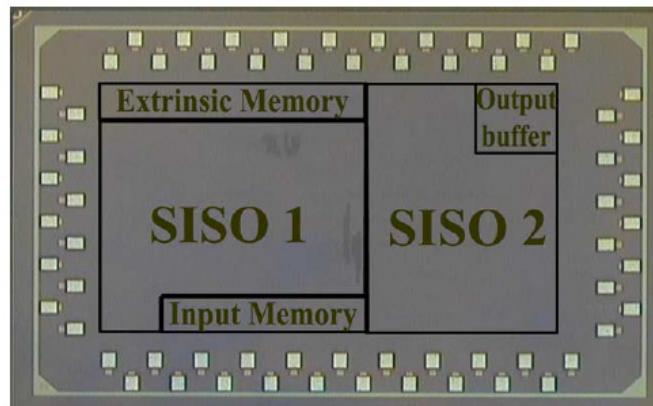
- Reciprocal dual trellis has shorter trellis stage and simpler structure

□ Example :



Dual Trellis Turbo Decoder Design (2/3)

- “A 40nm 535Mb/s Multiple Code-Rate Turbo Decoder Chip using Reciprocal Dual Trellis,” invited by *IEEE JSSC 2013*

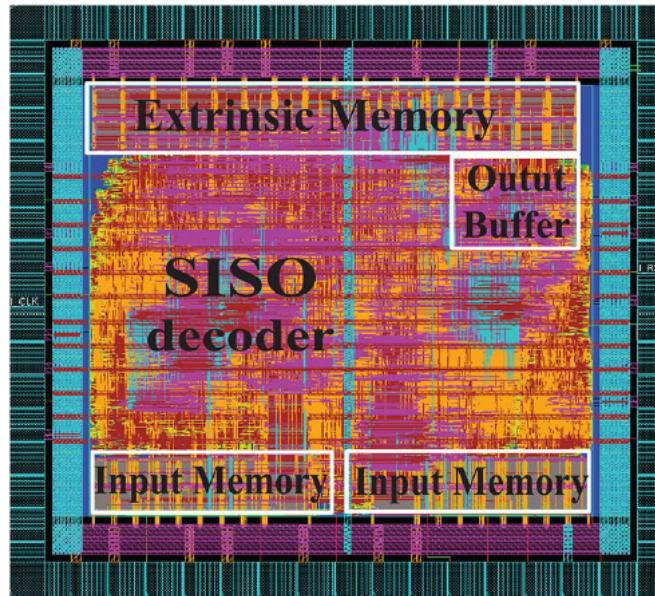


	Proposed
Technology	40 nm
Block size	4096
Supply voltage (V)	0.9
Code rate	1/3, 1/2, 2/3, 4/5, 8/9
Operation frequency (MHz)	252
Max. throughput (Mbps)	535 at 6 iter.
Core area (mm ²)	1.27
Gate counts (k)	900
Memory (kb)	152
Power consumption (mW)	218
Area efficiency (Mbps/k-gates) ^a	0.594
Energy efficiency (nJ/bit/iter) ^a	0.068
Status	Measurement

^a Throughput normalization to 6 iteration

Dual Trellis Turbo Decoder Design (3/3)

- “An Area Efficient Radix-4 Reciprocal Dual Trellis Architecture for a High Code-Rate Turbo Decoder,” *IEEE T-CAS II*, Jan. 2015.

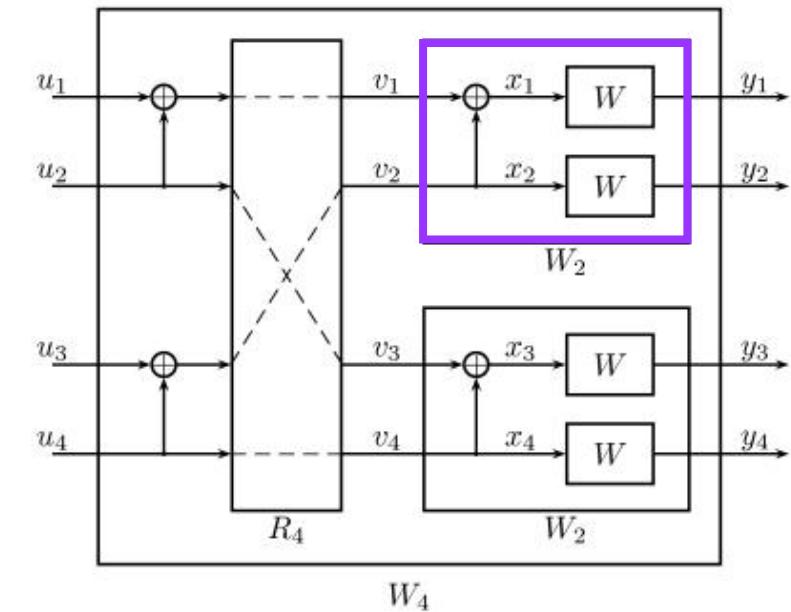
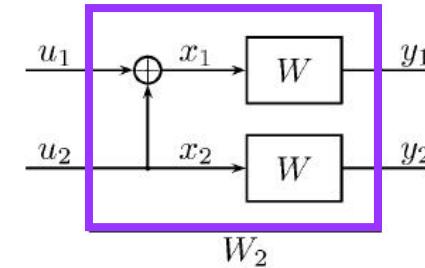


	Design	[10]	[15]	[7]
Technology (S nm)	90	40	65	65
Voltage (V)	0.9	0.9	1.2	1.1
Max. size	4096	4096	6144	6144
Radix	4	2	4	2
Max. code rate	8/9	8/9	0.95	0.95
Frequency (MHz)	185/370	252	410	300
Max. throughput @ 6 iter. (Mbps)	425	535	929	166
Core area (mm^2)	3.78	1.27	2.49	2.1
Gate counts (k)	600	900	1574	N/A
Memory (kb)	152	152	201	N/A
Power (mW)	310	218	966	300
Area efficiency (Mbps/k-gates)	0.71	0.59 ^a	0.59 ^a	N/A
Energy efficiency (nJ/bit/iter) ^a	0.122	0.068 ^a	0.17 ^a	0.31 ^a

^a: Normalize to 90 nm process; delay: 1/S; energy efficiency: $(1/VS)^2$

Polar Codes (1/4)

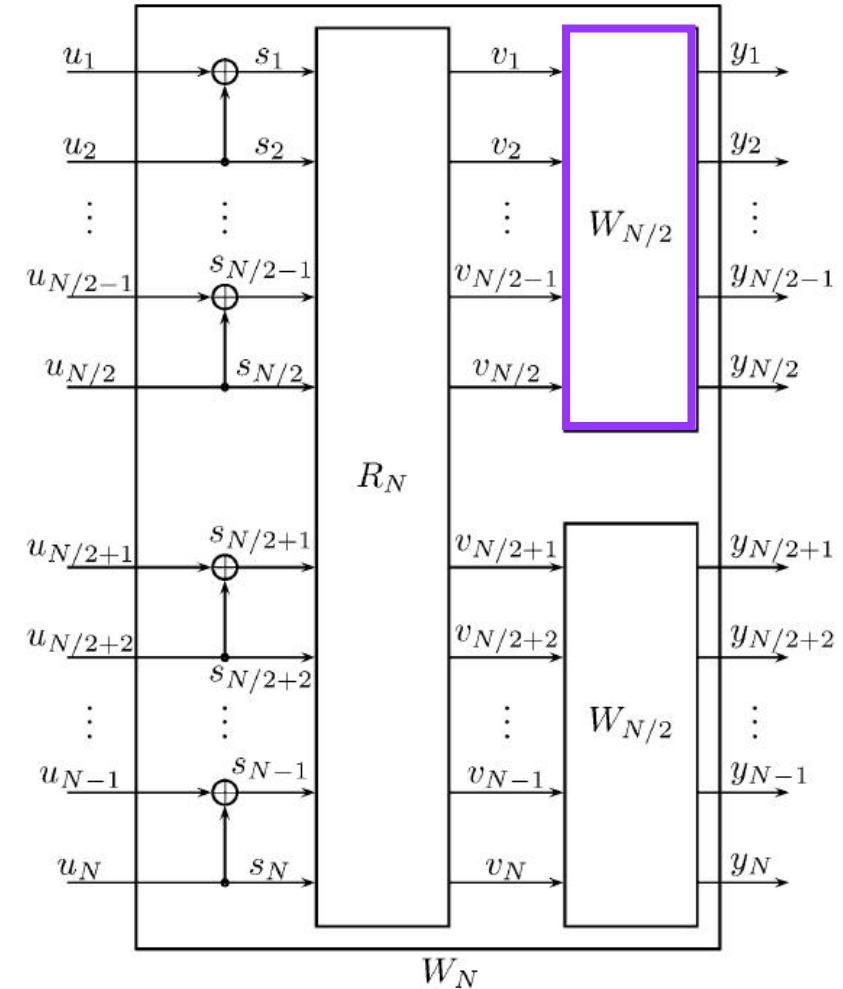
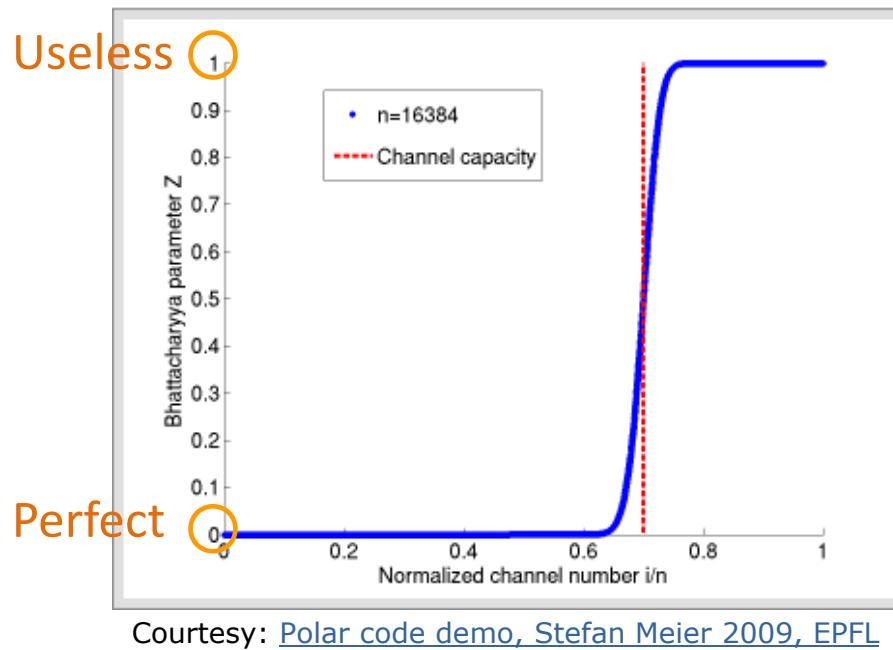
- New idea of ECC, proposed by Erdal Arikan in 2009
- A capacity-achieving code for BI-DMC
- Low complexity for both encoder & decoder
- Low error floor
- “Polarize” channels to better and worse ones



Polar Codes (2/4)

□ Channel Polarization

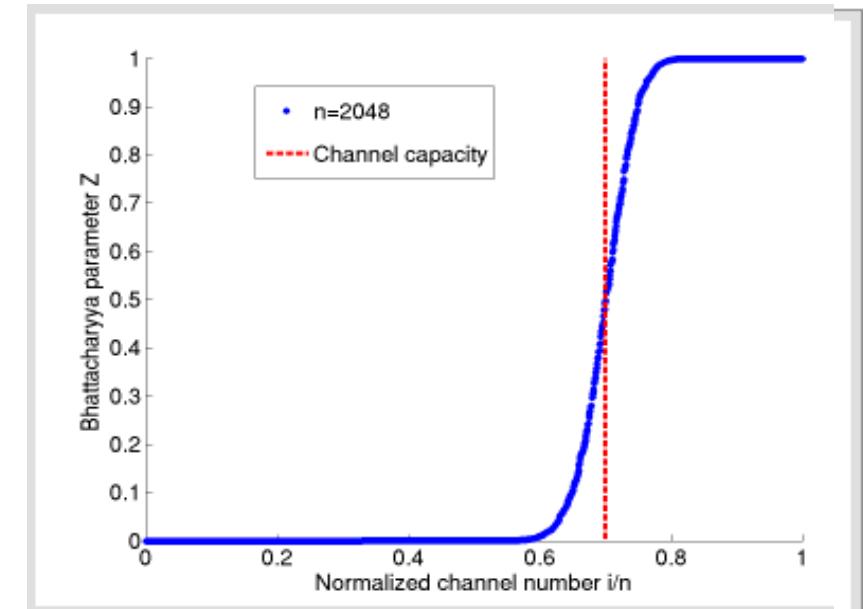
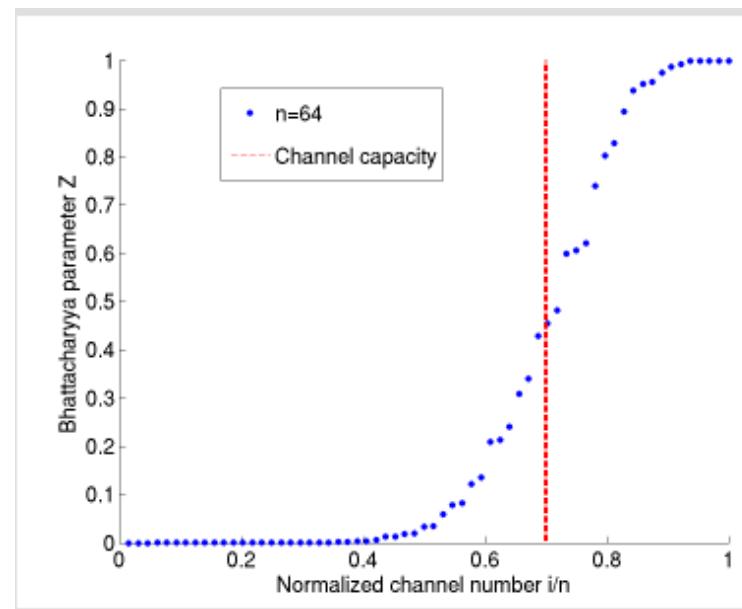
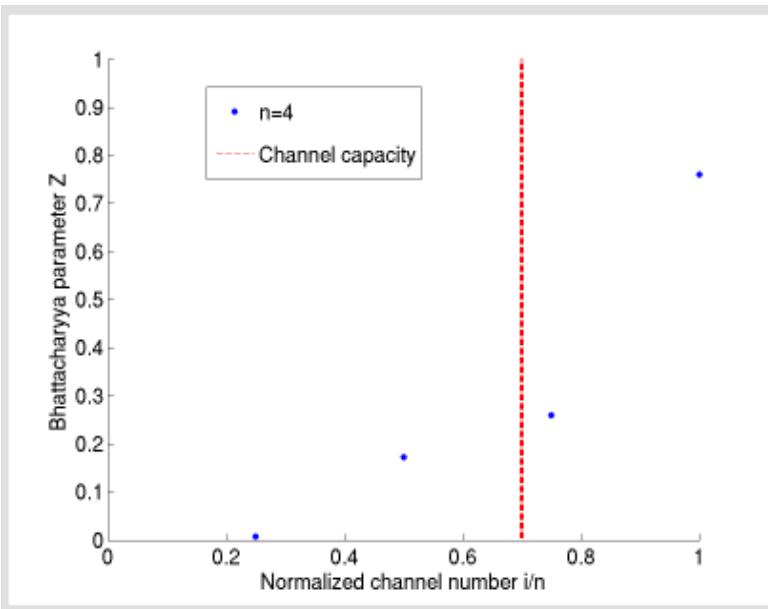
- After multiple times of polarization, the bit-channels tend to either useless or perfect



Polar Codes (3/4)

□ Speed of polarization

- Achieving channel capacity (Shannon's limit) as length N tends to infinity
- The percentage of better bit-channel depends on channel capacity of raw channel

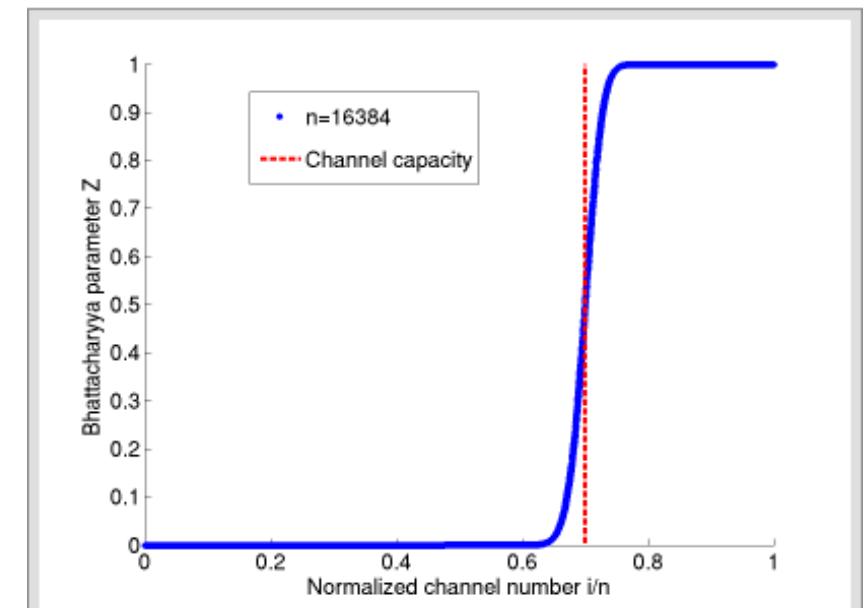
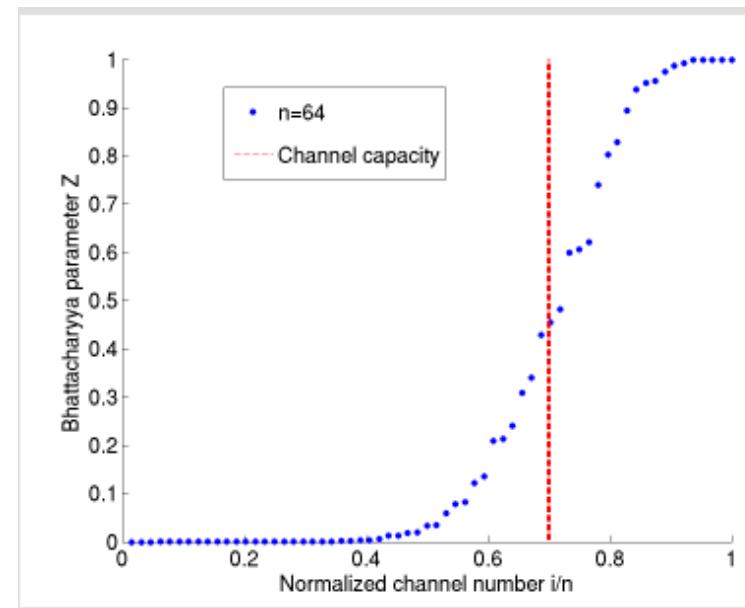
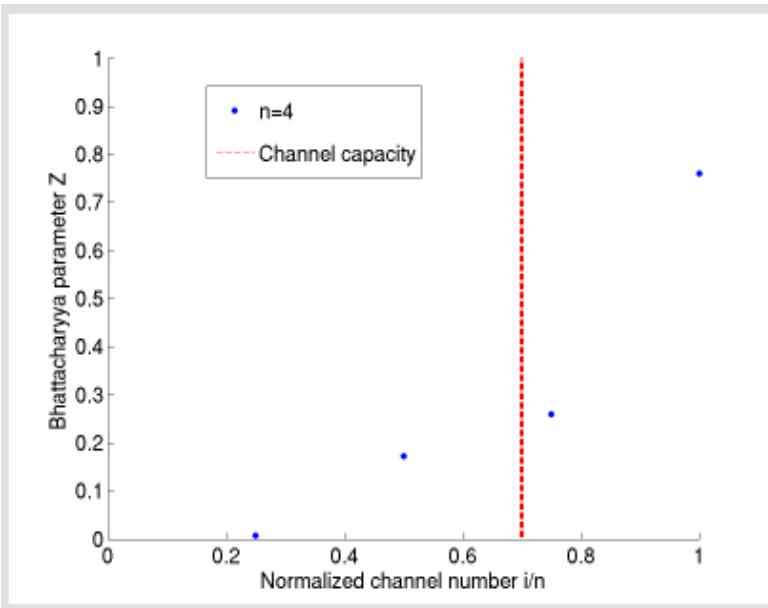


Courtesy: [Polar code demo, Stefan Meier 2009, EPFL](#)

Polar Codes (3/4)

□ Speed of polarization

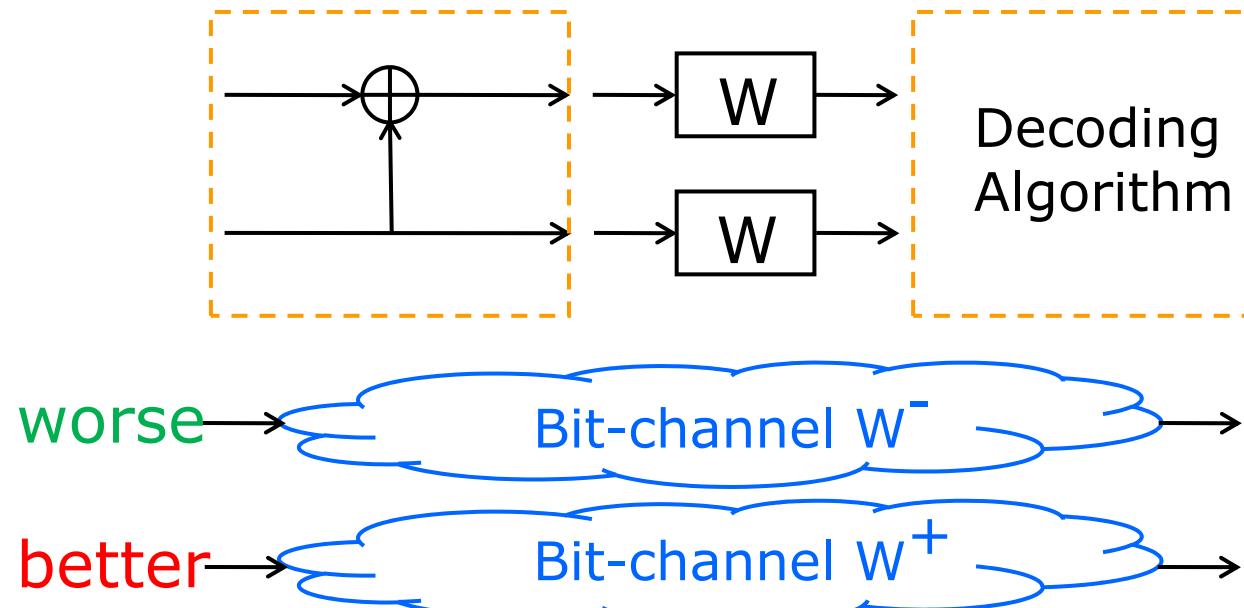
- Achieving channel capacity (Shannon's limit) as length N tends to infinity
- The percentage of better bit-channel depends on channel capacity of raw channel



Courtesy: [Polar code demo, Stefan Meier 2009, EPFL](#)

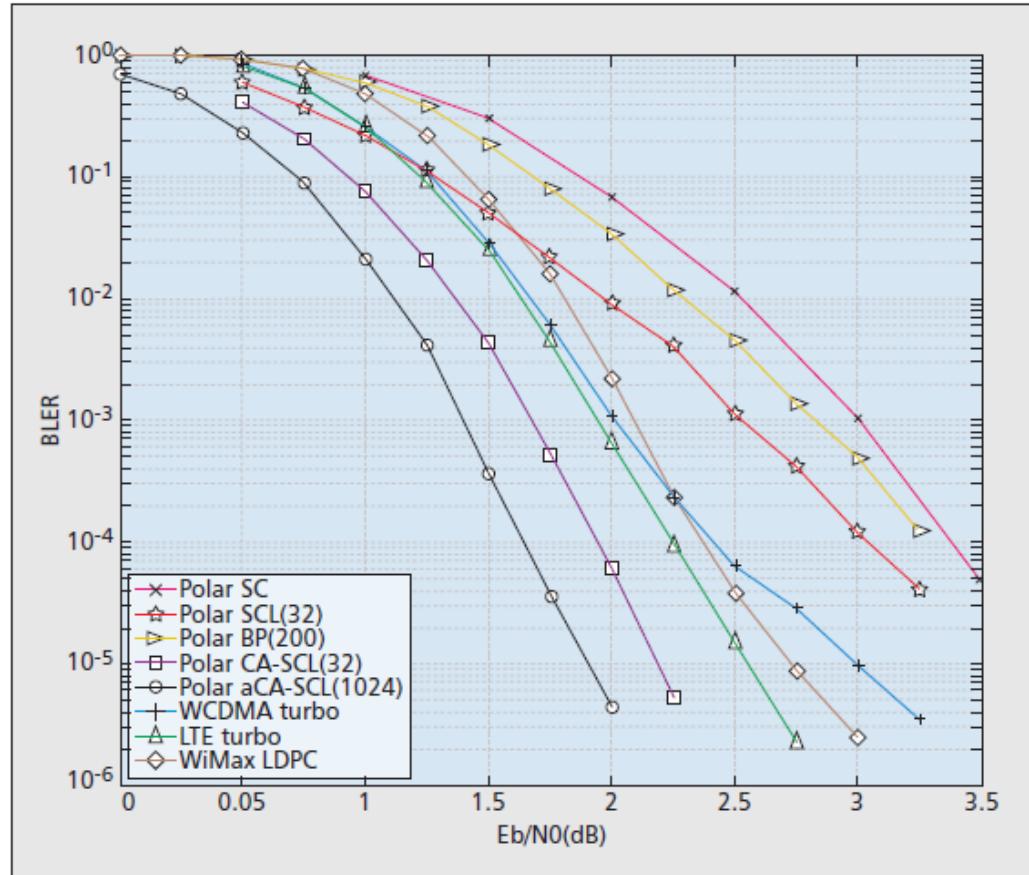
Polar Codes (4/4)

- Only putting information bit in **better** bit-channels
- On the other hand, putting frozen bit, which is a known value for both transmitter and receiver, in **worse** bit-channels



Decoding Performance Comparison (1/2)

- (1024,512) for all codes, except for WiMax LDPC which is (1056,528)



SC: successive cancellation decoding

SCL: list successive cancellation decoding

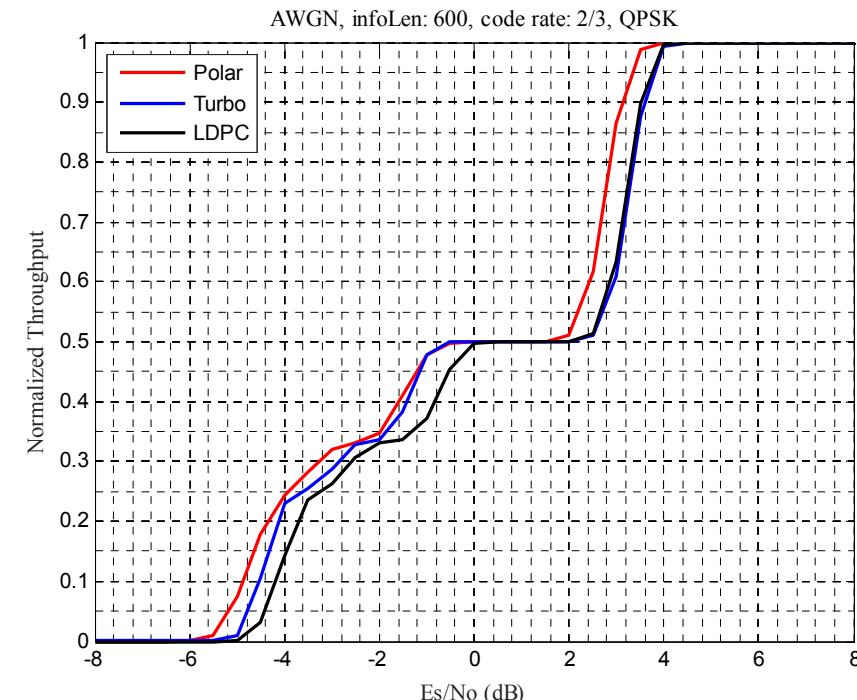
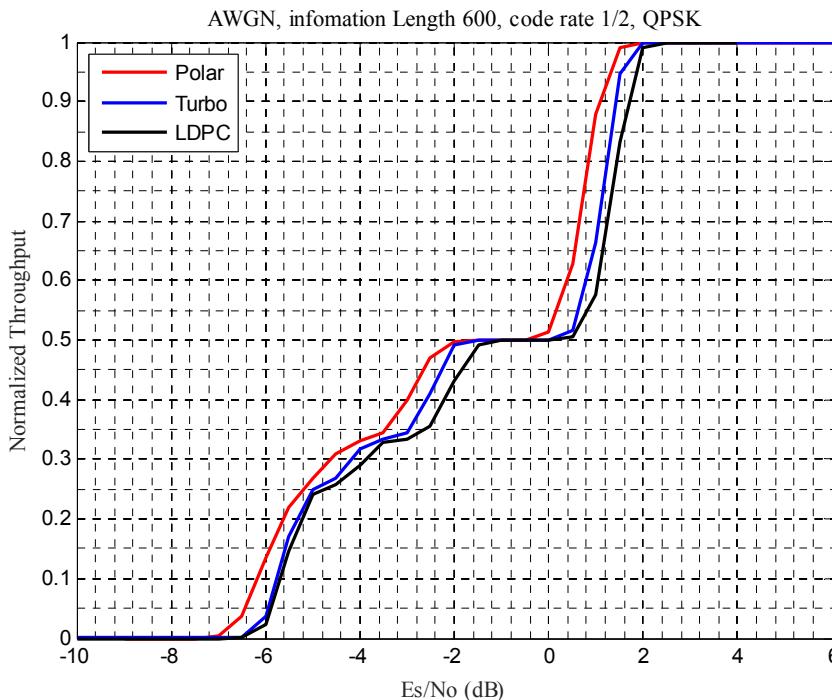
CA-SCL: CRC-aided SCL

aCA-SCL: adaptive CRC-aided SCL

Decoding Performance Comparison (2/2)

□ 3GPP TDocs (written contributions) at meeting

- <http://www.3gpp.org/DynaReport/TDocExMtg--R1-86--31663.htm>
- **In Huawei's result, the performance of polar codes are better than turbo and LDPC codes for small information bit size (e.g., <1000)**



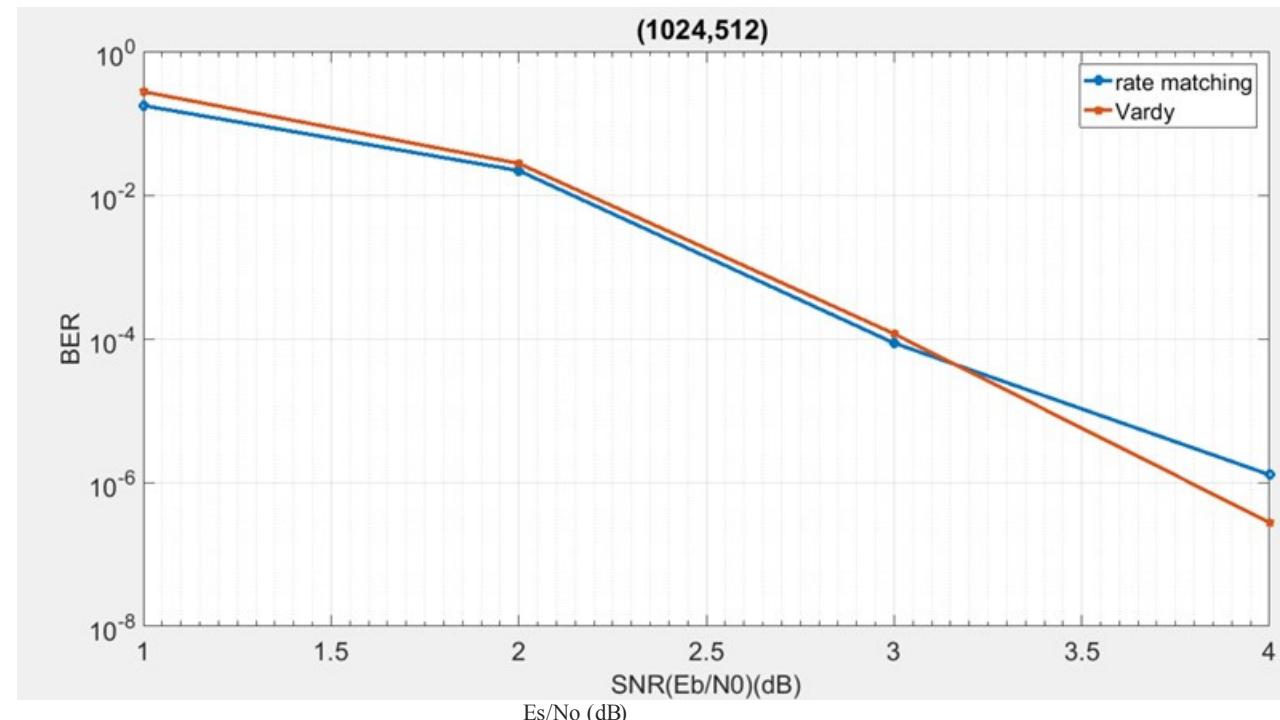
Decoding Performance Comparison (2/2)

□ 3GPP TDocs (written contributions) at meeting

- <http://www.3gpp.org/DynaReport/TDocExMtg--R1-86--31663.htm>
- In Huawei's result, the performance of polar codes are better than turbo and LDPC codes for small information bit size (e.g., <1000)

Assume $i \triangleq B_{n-1}B_{n-2}\dots B_0$ with $B_j \in \{0,1\}, j = [0,1,\dots,n-1]$,
then,

$$W_i = \sum_{j=0}^{n-1} B_j * 2^{j*\frac{1}{4}}$$



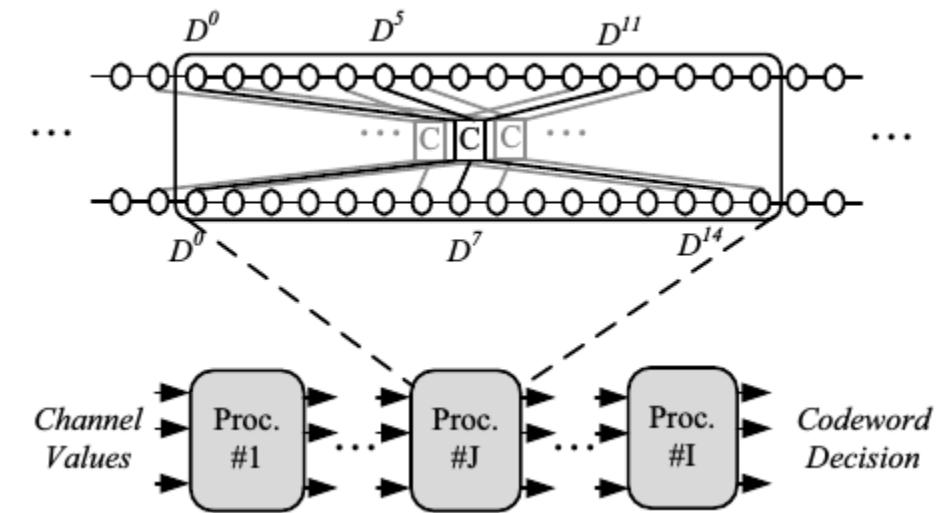
Low-Density Parity Check Convolutional Codes (1/2)

- A Semi-infinite and time-varying H
- The same message-passing decoding
 - Sliding window performed along the received codewords
 - Variable nodes that are $(ms+1)$ apart can be updated concurrently
 - $(ms+1)$ decoding process are performed concurrently
- Time-invariant representation as convolutional code,

$$\mathbf{H}_0^T(D) = \begin{bmatrix} 1 + D^5 + D^{11} \\ 1 + D^7 + D^{14} \end{bmatrix}$$

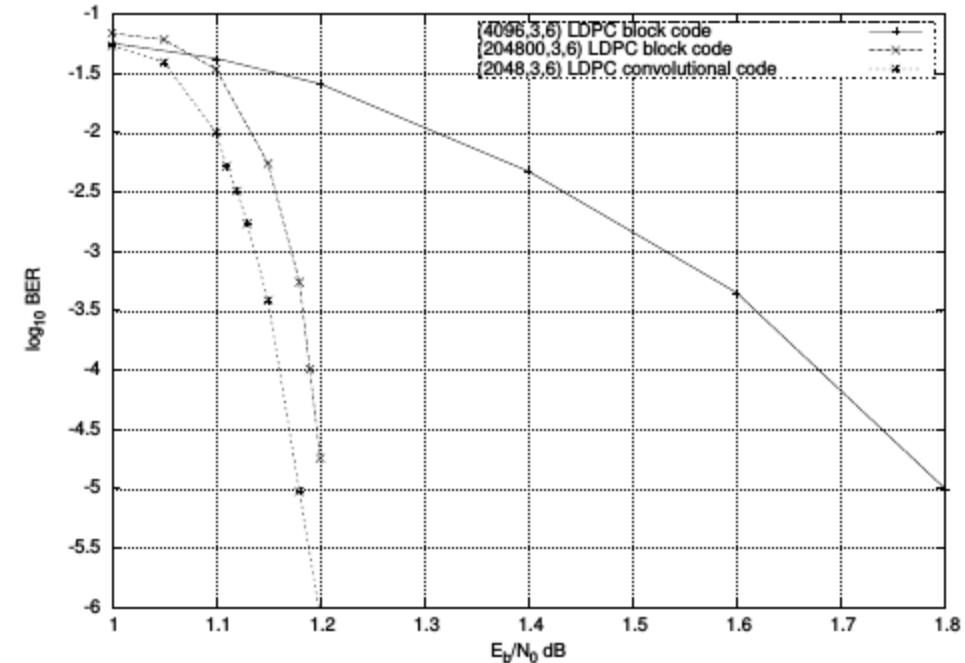
Example of $R = 1/2, (14,3,6)$ decoder

$$\mathbf{H}^T = \begin{bmatrix} \mathbf{H}_0^T(0) & \Lambda & \mathbf{H}_{m_s}^T(m_s) \\ O & O & O \\ & \mathbf{H}_0^T(t) & \Lambda & \mathbf{H}_{m_s}^T(t+m_s) \\ & O & O & O \end{bmatrix}$$



Low-Density Parity Check Convolutional Codes (2/2)

- The time-separable Tanner graphs facilitates high-clock rate and fully parallelism implementation
 - Relax the routing congestion issue
 - More amenable to pipeline architecture
- Flexible code length/rate designs
- **Suitable for streaming data service**



Comparison of terminated LDPC-CC and LDPC-BC ($R = 1/2$) *

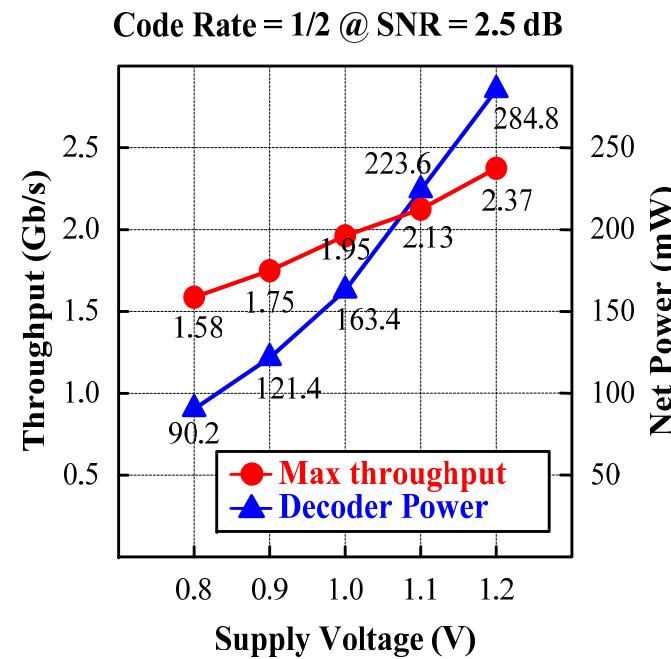
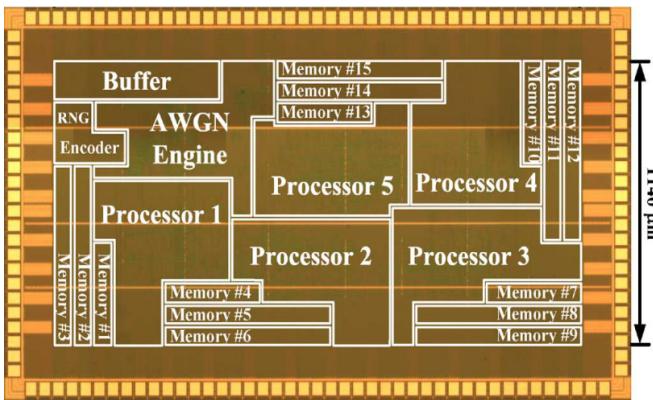
* Costello Jr, Daniel J., et al. "A comparison between LDPC block and convolutional codes." Proc. Information Theory and Applications Workshop. 2006.

Low-Density Parity Check Convolutional Codes (2/2)

	LDPC-BC	LDPC-CC	Turbo Code
Variable Frame Size	No	Yes	Yes
Multiple Code Rates	Depends	Yes	Yes
Simple Encoder	Depends	Yes	Yes
Routing of Decoder	Complicated	Simple	Simple
Parallelization of Decoder	Very High	High	Medium
Throughput	Over 10 Gbps	~ 5 Gbps	~ 1 Gbps
Hardware Cost	Medium	Medium	High
Energy Efficiency (nJ/bit/iter.)	10~0.01	1~0.01	10~0.1
Error-Correcting Capability	Excellent	Excellent	Extraordinary

LDPC-CC Decoder Design (1/3)

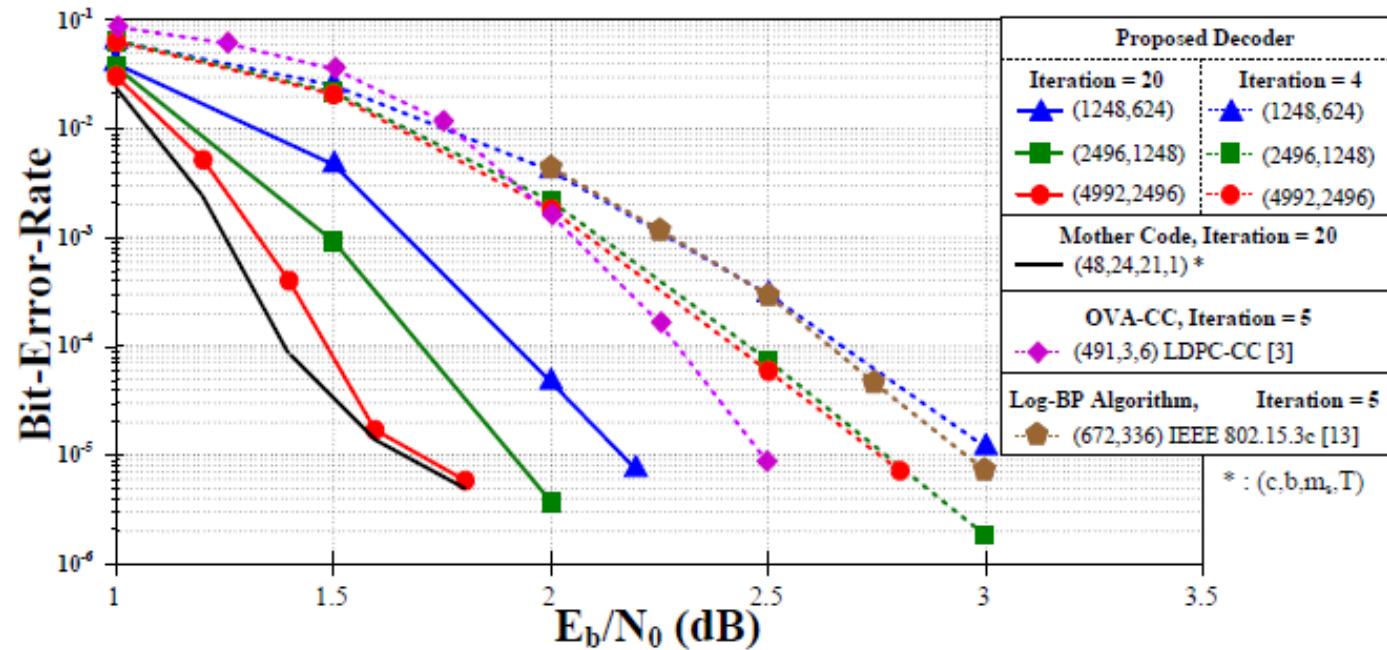
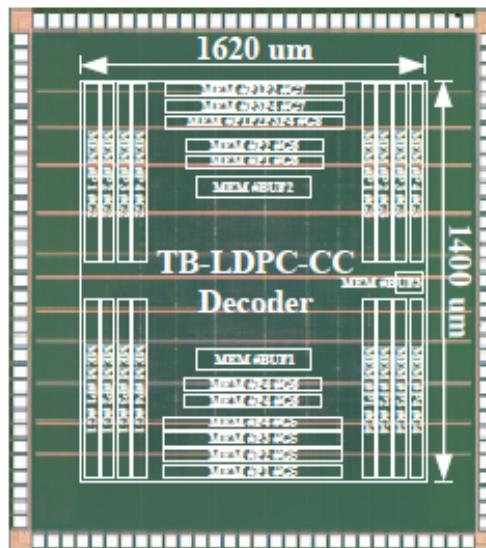
- A 2.37Gb/s 284.8mW Rate-Compatible (491,3,6) LDPC-CC Decoder
 - Presented in *VLSI Symposium 2011*, Invited by *IEEE JSSC 2012*



Process	UMC 90nm 1P9M
Constraint Length	984 (=491+1)*2
Code Type	Time-Varying (T=3)
Code Rate	1/2, 2/3, 3/4, 4/5, 5/6
Input Quantization	6 bits
Processor Number	5
Parallelization Factor	12
Max. Throughput	2.37 Gbps @ 198MHz
Total Memory Size	52.5 kbits
Decoder Area	2.24 mm ²
Utilization	87.8 %
Power	284 mW (@1.2 V)
Energy Efficiency	0.024 nJ/bit (@1.2V)

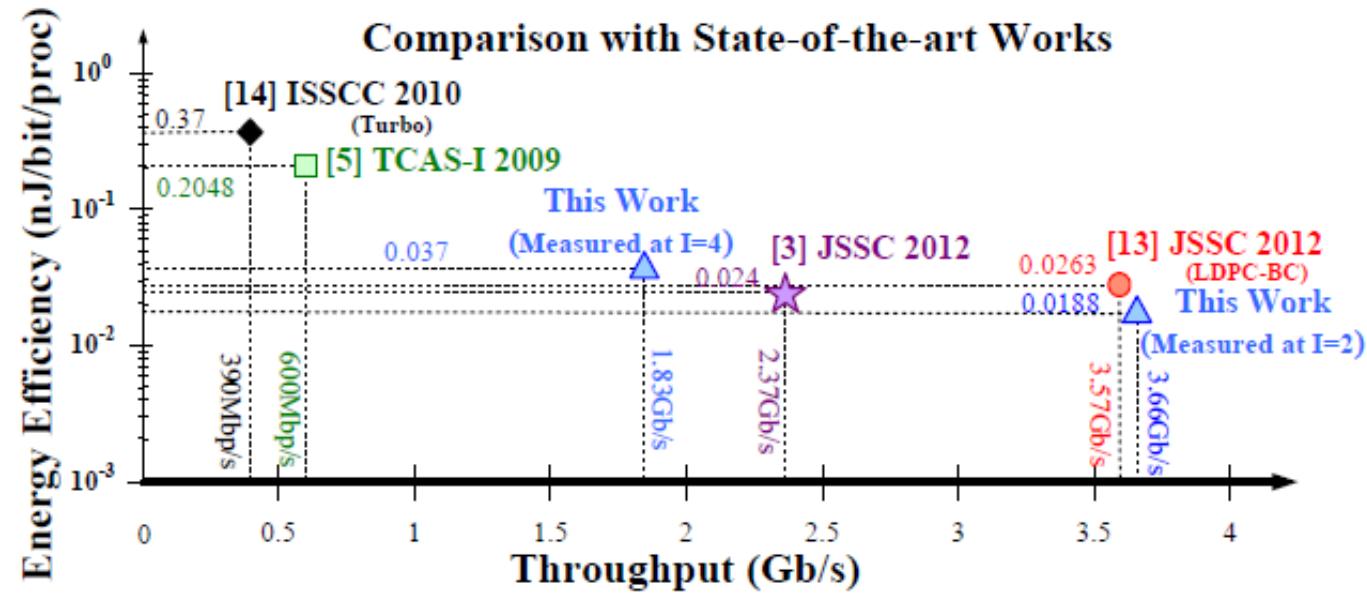
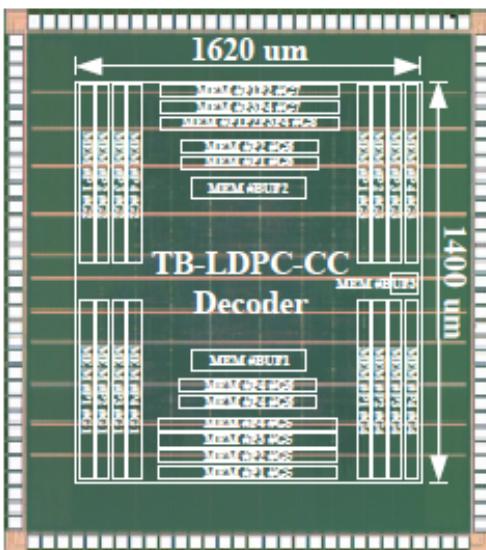
LDPC-CC Decoder Design (2/3)

- “A 3.66Gb/s 275mW TB-LDPC-CC Decoder Chip for MIMO Broadcasting Communications”, presented in *IEEE A-SSCC 2013*



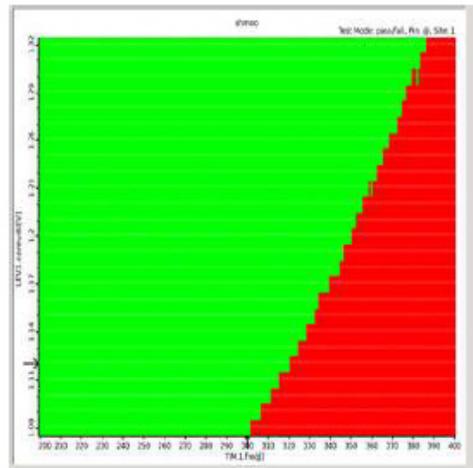
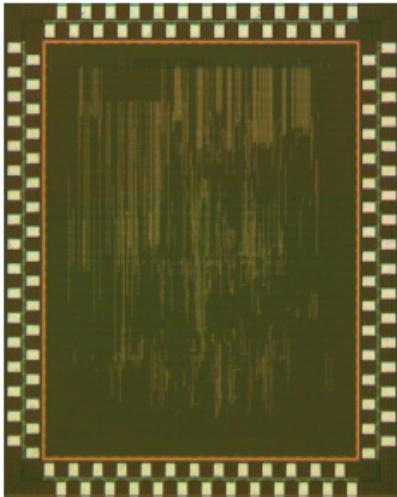
LDPC-CC Decoder Design (2/3)

- “A 3.66Gb/s 275mW TB-LDPC-CC Decoder Chip for MIMO Broadcasting Communications”, presented in *IEEE A-SSCC 2013*



LDPC-CC Decoder Design (3/3)

- “A 7.72Gb/s LDPC-CC Decoder with Overlapped Architecture for Pre-5G Wireless Communications”, presented in *IEEE ASSCC 2016*



- [4] C. L. Chen, Y. H. Lin, H. C. Chang, and C. Y. Lee, “A 2.37-Gb/s 284.8 mW rate-compatible (491,3,6) LDPC-CC decoder,” *IEEE J. Solid-State Circuits*, vol. 47, no. 4, pp. 817–831, 2012.
- [5] Y. Chen, Q. Zhang, D. Wu, C. Zhou, and X. Zeng, “An efficient multirate LDPC-CC decoder with a layered decoding algorithm for the IEEE 1901 standard,” *IEEE Trans. Circuits Syst. II*, vol. 61, no. 12, pp. 992–996, Dec 2014.

COMPARISON OF PREVIOUS DESIGNS

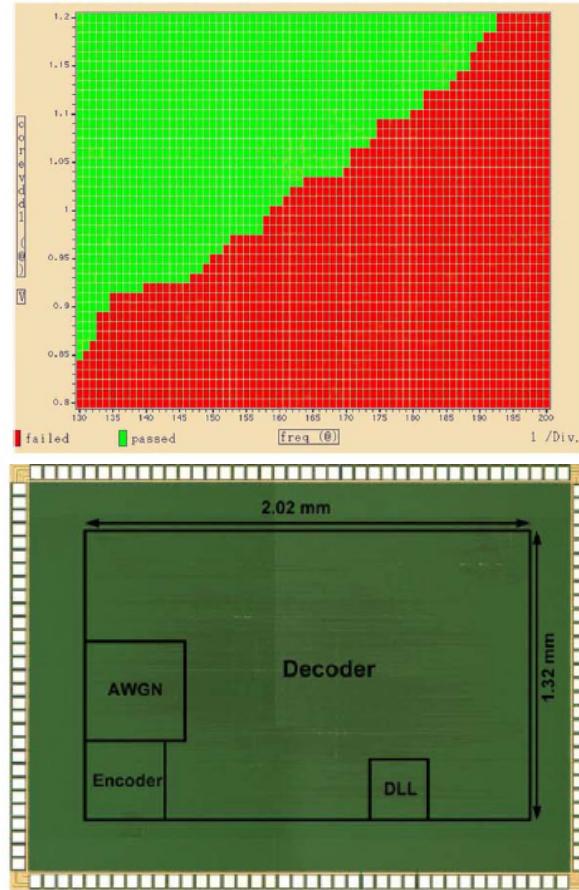
	This work	JSSC'12 [4]	TCAS-II'14 [5]
Code	LDPC-CC	LDPC-CC	LDPC-CC
Code Rate	4 rates	5 rates	4 rates
Constraint	1135	984	1135
Length Tech.	65nm	90nm	130nm
Quantization	6-bit	6-bit	6-bit
# Proc.	6	5	10
Memory(Kb)	47.04	52.5	152.76
Area(mm^2)	1.19(0.198)	2.24(0.269)	3.55(0.089)
Max Frequency	322 MHz	198 MHz	180 MHz
Data Rate(Gbps)	7.728(7.728)	2.37(2.95)	0.3(0.54)
Power	410.5 mW	284 mW	200.4 mW
Energy Efficiency ^a	8.75	24.0	66.8
Area	39.03	5.29	0.85
Efficiency ^b	(39.03)	(10.97)	(6.06)

a : pJ/bit/Processor #

b : Throughput / (Area/Processor #)

(): Technology normalized

Stochastic LDPC Decoder Design



	This Work	JSSC'12 [7]	IEICE'11 [6]
CMOS Technology	90-nm	65-nm	65-nm
Code Length	672		
Code Rate [Standard]	1/2, 5/8, 3/4, 7/8 [802.15.3c]		
Clock Frequency (MHz)	768 / 560 ¹	197 (142)	400 (289)
Gate Count (K)	760.3	647	430
Core Area (mm^2)	2.67	1.56 (2.99)	1.30 (2.49)
Data Rate (Gb/s)	7.92 / 5.78 ¹	5.79 (4.18)	5.88 (4.25)
Power (mW)	437.2 / 280.7 ¹	361 (260.7)	537.6 (388.3)
Chip Density (%)	90	73.3	86.3
Energy Efficiency (pJ/bit)	55.2 / 48.6 ¹	62.4	91.4
Hardware Efficiency ³ (Mb/s/mm ²)	5.04 / 3.68 ¹	(2.38)	(2.9)
Algorithm	Stochastic	NMS	NMS
Status	Measurement	Measurement	Measurement

⁰: normalized to 90-nm technology, 1 V supply

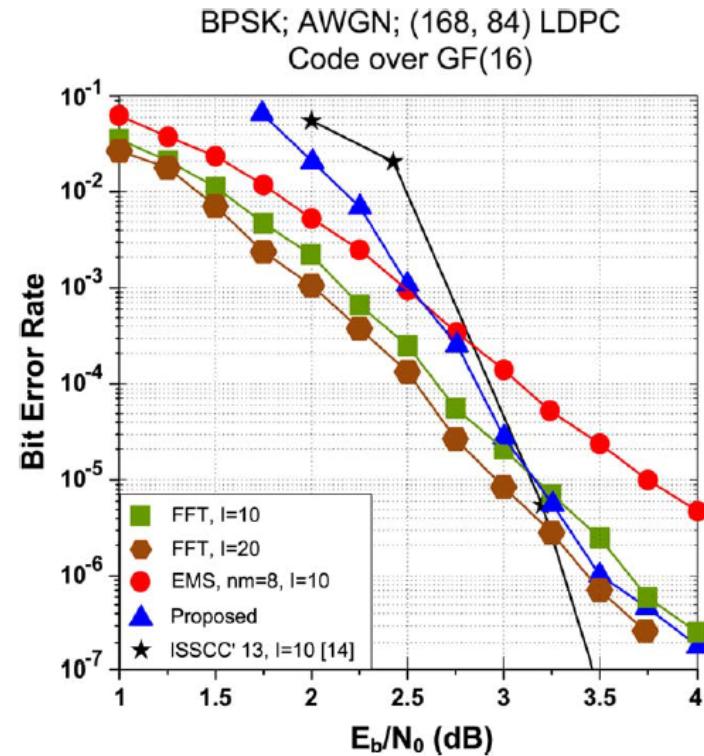
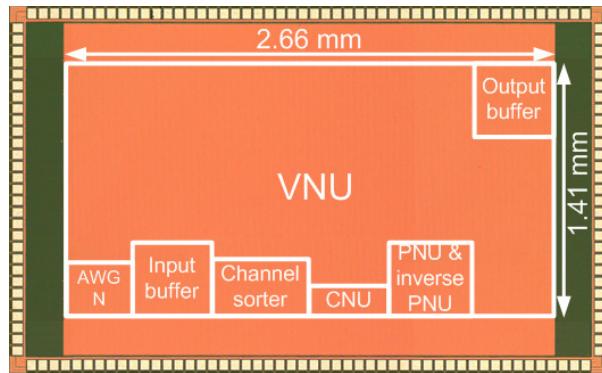
¹ Measured at 1 V supply

² BER is below 10^{-14}

³ Scaled data rate per information bit per scaled area

- “A 7.92 Gb/s 437.2 mW **Stochastic LDPC Decoder Chip** for IEEE 802.15.3c Applications,” *IEEE T-CAS I*, Feb. 2015.

NB-LDPC-BC Decoder Design (1/2)

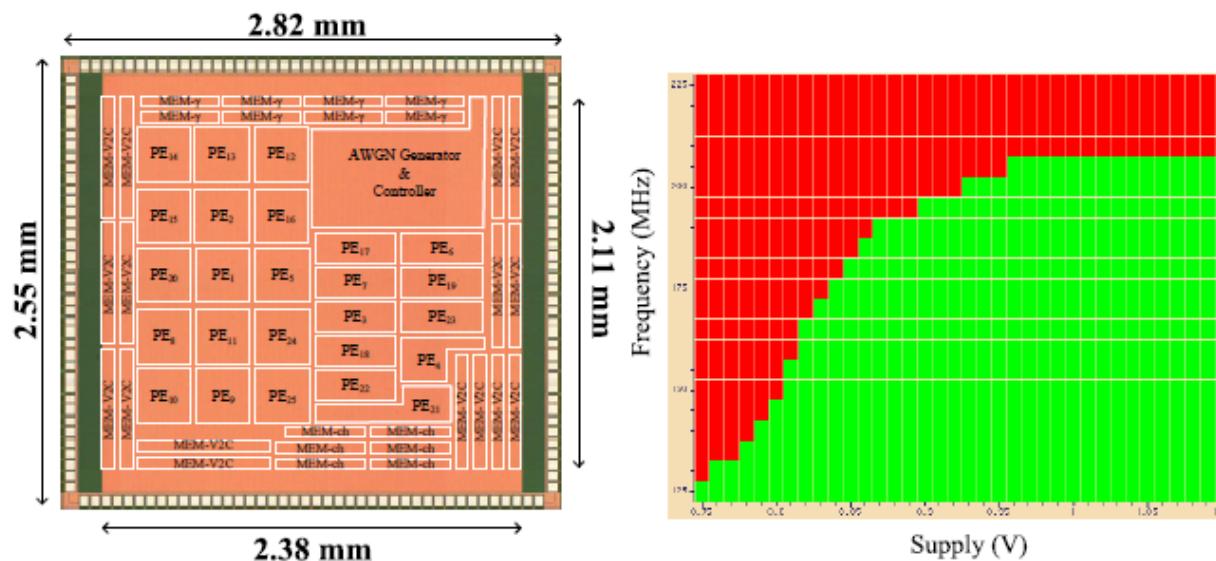


	This work	JSSC' 15 [3]
Code	(168, 84) over GF(16)	(160, 80) over GF(64)
Block length	672	960
Process (nm)	90	65
Gate count (K-gate)	1103	2780
Core Area (mm^2)	3.75	7.04
Frequency (MHz)	368	700
Throughput (Mb/s)	1315	1221
Power (mW)	587.7	3704
Area efficiency ($Mb/s/mm^2$)	350.67	173.44
Energy efficiency (nJ/bit)	0.45	3.034
Algorithm	Stochastic	EMS
Status	measurement	measurement

- “A 1.31Gb/s, 96.6% Utilization Stochastic **Non-binary LDPC Decoder** for Small Cell Applications”, *IEEE ESSCIRC 2015*.

NB-LDPC-BC Decoder Design (2/2)

- “An Efficient Decoder Architecture for NB-LDPC Codes with Extended Min-Sum Algorithm,” *IEEE T-CAS II* 2016
- “A 90nm 1.53Gb/s 0.28nJ/b NB-LDPC Decoder with Message-Length Switching,” in preparation of journal submission



[11] Y. S. Park, Y. Tao, and Z. Zhang, “A fully parallel nonbinary LDPC decoder with fine-grained dynamic clock gating,” *IEEE Journal of Solid-State Circuits*, vol. 50, no. 2, pp. 464–475, Feb. 2015.

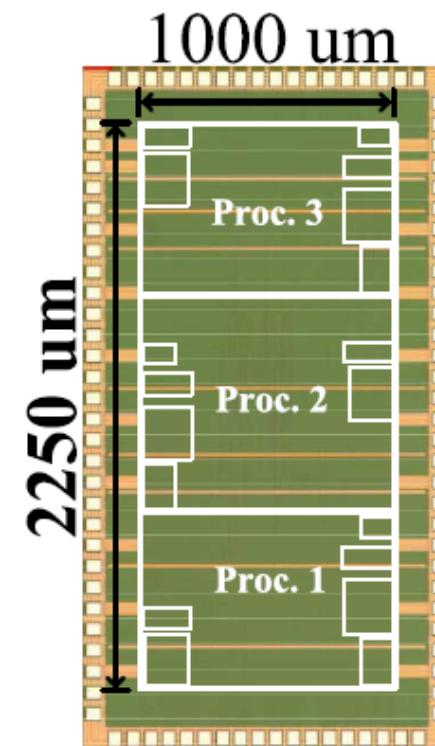
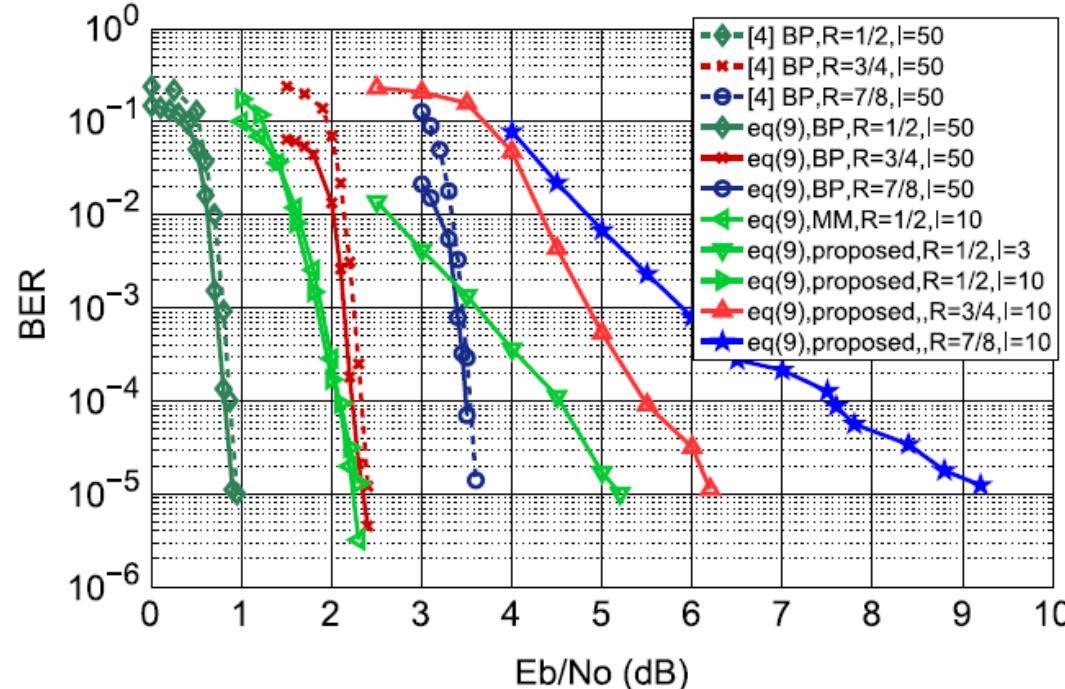
	This work Measurement	JSSC'15 [11] Measurement
Technology	90 nm	65 nm
Code Length(bit)	2000	960
Code Rate	0.5	0.5
Galois Field	GF(32)	GF(64)
Decoding Algorithm	T-MM	EMS
Iteration	1-10(avg. 4)	10-30(avg. 11.7)
Performance (dB)	2.2 ^a	3.0 ^a
Gate Count(M)	1.06	2.78
Clock Frequency(MHz)	200	700
Throughput(Mb/s)	1530	1150
Power(mW)	434	3870
Hardware Efficiency ^c	1443	414(299 [†])
Energy Efficiency(nJ/bit)	0.28	3.37
Energy Efficiency (nJ/bit/Iteration)	0.07	0.29(12.4 [†])

a : The SNR with BER=10⁻⁵

c : Throughput / Gate count (Mb/s/M Gate)

NB-LDPC-CC Decoder Design

- “A (50,2,4) NB-LDPC-CC Decoder Chip over GF(256) in 90nm CMOS,” presented in *IEEE A-SSCC 2012*.
- “*Joint Designed NB-LDPC-CC and Memory-Based Decoder Architecture*,” *IEEE T-CAS I 2015*

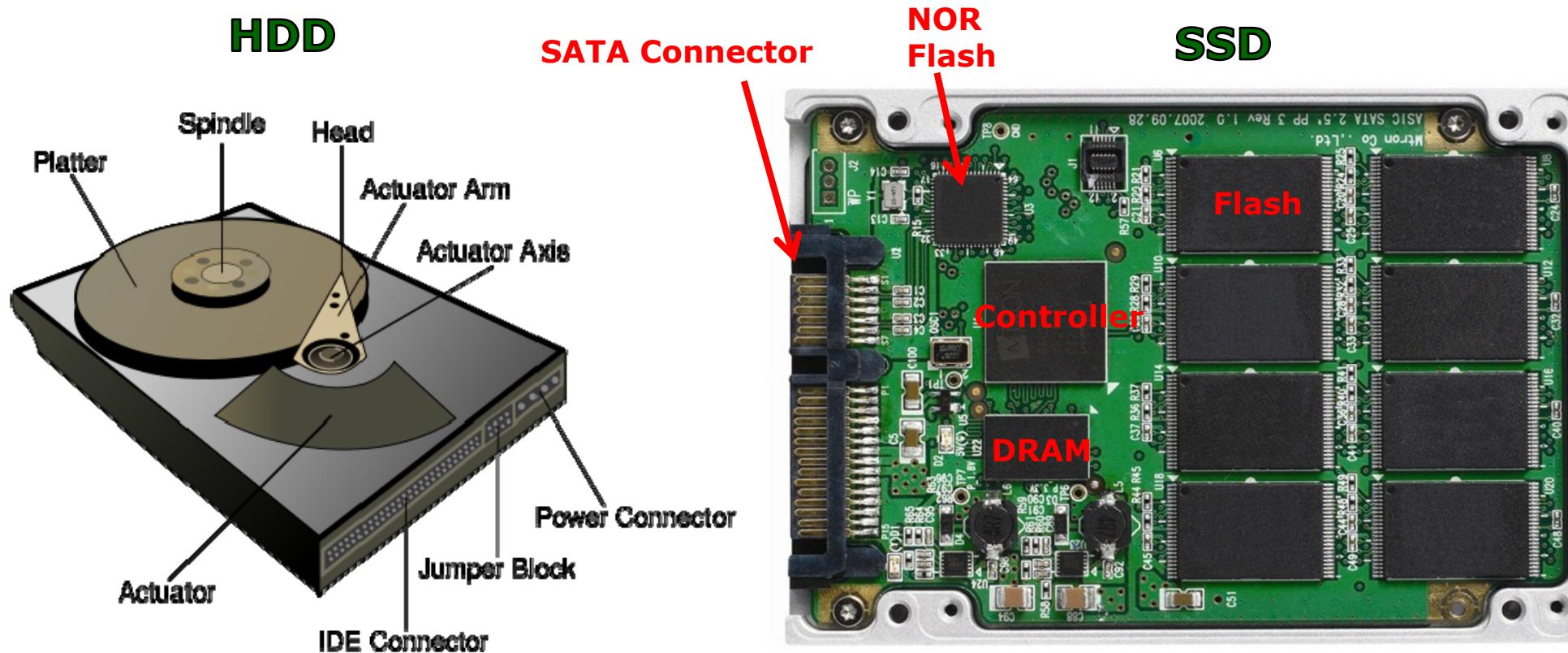


Outline

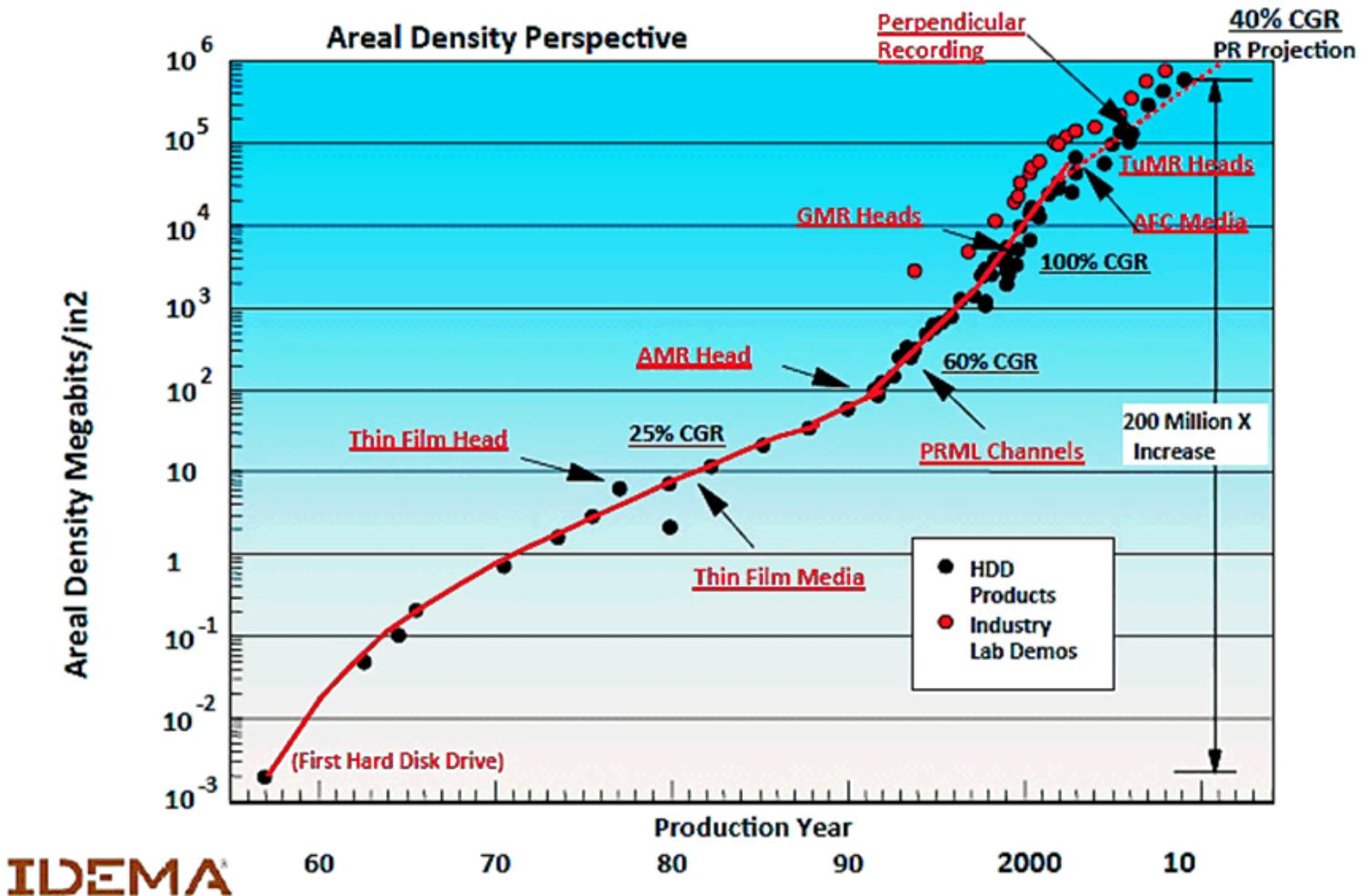
- ***Introduction***
 - *FEC = ECC + Randomization + Interleaving*
 - ***ECC in 5G Communications Systems***
 - *Turbo Codes vs. LDPC Codes*
 - *Polar Codes in short block length*
 - ***ECC in Non-volatile Memory (NVM) Systems***
 - *BCH Codes vs. LDPC Codes*
 - *Soft Information in NAND Flash Memory*
 - ***Summary***
 - ***Reference***
-

Let's talk about HDD firstly

- SSD uses non-volatile solid state memory (NAND Flash) to retain data and emulates HDD in many applications.

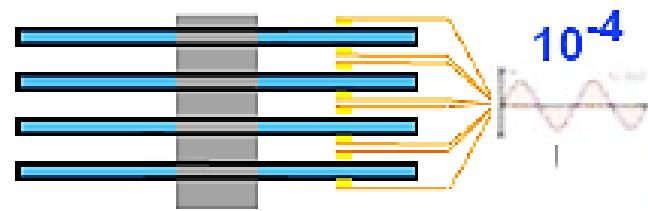


How to turn BAD into GOOD??



The Answer is:

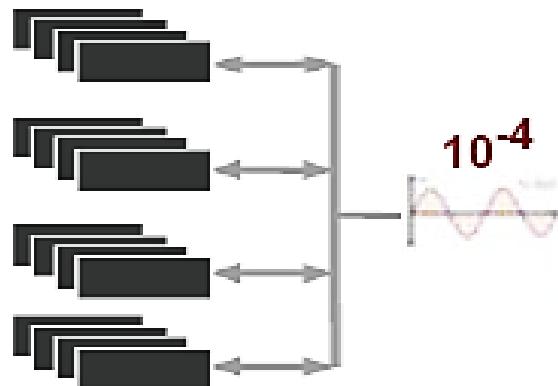
HDD Media



**Adaptive Signal Processing for Media Rd/Wr/Erase
Advanced Bit Detection & Error Correction Codes
Defect Management**

10^{-16}

Flash Media



**Adaptive Signal Conditioning for Flash Media
Auto Bit Detection & Error Correction Codes
Defect Management**

10^{-17}

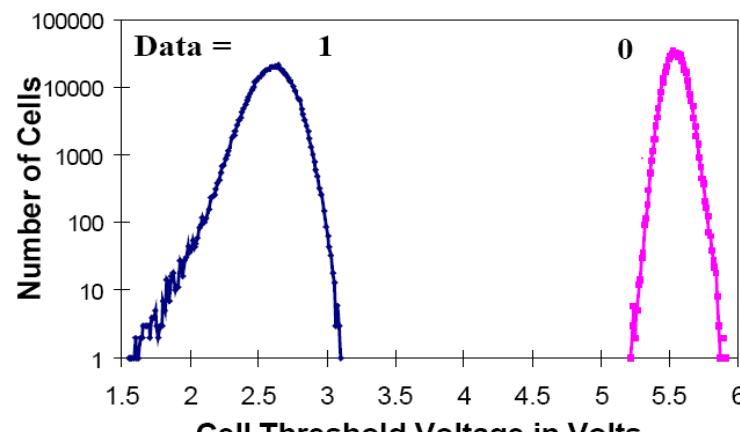
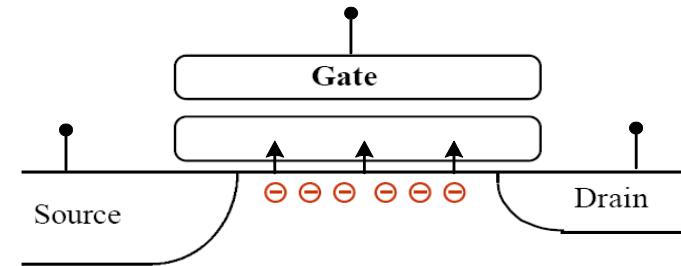
Source: IMEX research



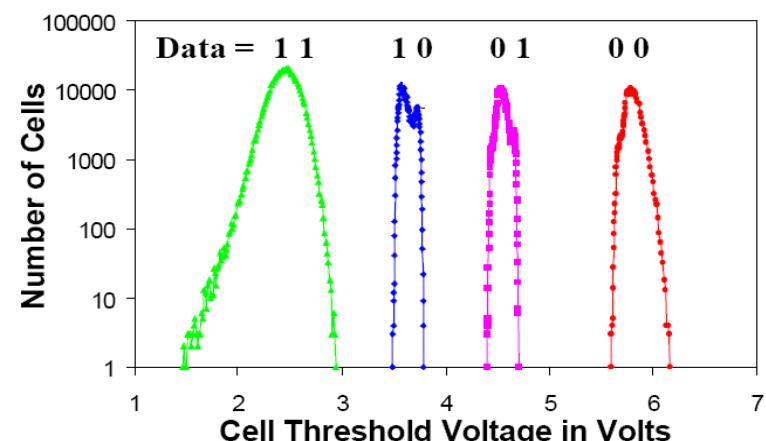
ECC + Signal Processing

Non-Volatile Memory

- ROM、EPROM、EEPROM、PCM、MRAM、RRAM
- Flash memory
 - Floating Gate
 - Single Level Cell / Multi Level Cell
 - Threshold voltage distribution



SLC (Single Level Cell)



MLC (Multiple Levels Cell)

ECC in NOR Flash Memory (1/2)

- BCH codes
 - Found by Hocquenghem (1959), Bose (1960) and Chaudhuri (1960)
 - Reed-Solomon code is a non-binary BCH code

- Consider a t -error-correcting BCH code over $GF(2^4)$

- Let α is a primitive element of $GF(2^4)$ with $p(x) = x^4 + x + 1$
- The **minimum polynomials** of α and α^3 are

$$m_1(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^4)(x - \alpha^8) = x^4 + x + 1$$

$$m_3(x) = (x - \alpha^3)(x - \alpha^6)(x - \alpha^{12})(x - \alpha^9) = x^4 + x^3 + x^2 + x + 1$$

- For **single**-error-correcting, $g_{SEC}(x) = m_1(x) = x^4 + x + 1$

- For **double**-error-correcting, $g_{DEC}(x) = LCM\{m_1(x), m_3(x)\}$

$$= m_1(x) \cdot m_3(x) = x^8 + x^7 + x^6 + x^4 + 1$$

ECC in NOR Flash Memory (2/2)

□ Fully Parallel BCH Codec with t=2 and t=3

	Design I [1]	Design II	CICC 09' [2]
	Codec	Codec	Decoder
(n,k)	(274,256)	(283,256)	(274,256)
Correctability	2 bits	3 bits	2 bits
Technology	90 nm	90 nm	180 nm
Latency	2.5 ns	4.0 ns	4.51 ns
Gate Count	12.5K	27.5K	28.3K

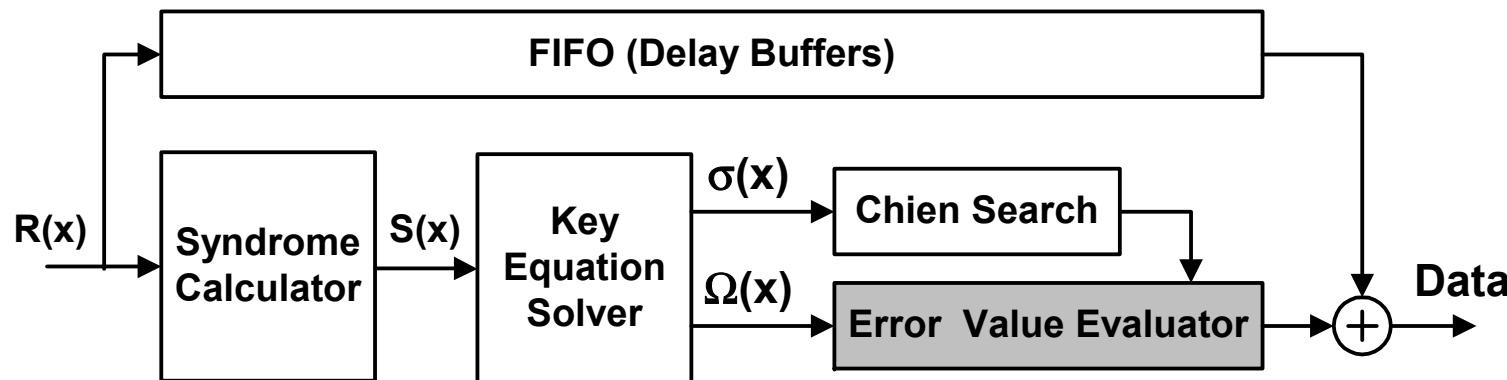


Chip Photo

- [1] “A fully parallel BCH codec with double error correcting capability for NOR flash applications,” *ICASSP 2012*.
[2] “Embedded high-speed BCH decoder for new-generation NOR flash memories,” in *CICC 2009*.

Systematic Approach for BCH/RS Decoders

- While dealing with VLSI design methodology, improvements for Cost, Speed, and Power are mutually exclusive.
- In RS (or BCH) decoding, there are 4 (or 3)-step decoding procedures
 - Calculating the syndrome polynomial $S(x)$
 - Solving the **Key Equation** ($\Omega(x) = \sigma(x)S(x) \bmod x^{2t}$)
 - Finding error locations by **Chien search**
 - Finding error values by **Forney algorithm**



ECC in NAND Flash Memory (1/3)

- A (522, 512) RS code was proposed by Toshiba
 - "A Compact On-Chip ECC for Low Cost Flash Memories", IEEE JSSC, 1997.

m	N	K	code rate
3	7	4	0.57
4	12	8	0.67
5	21	16	0.76
6	38	32	0.84
7	71	64	0.9
8	136	128	0.94
9	265	256	0.97
10	522	512	0.98

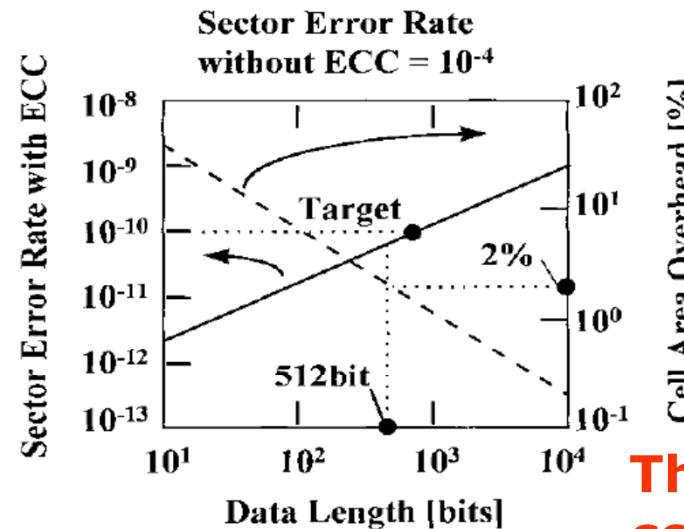
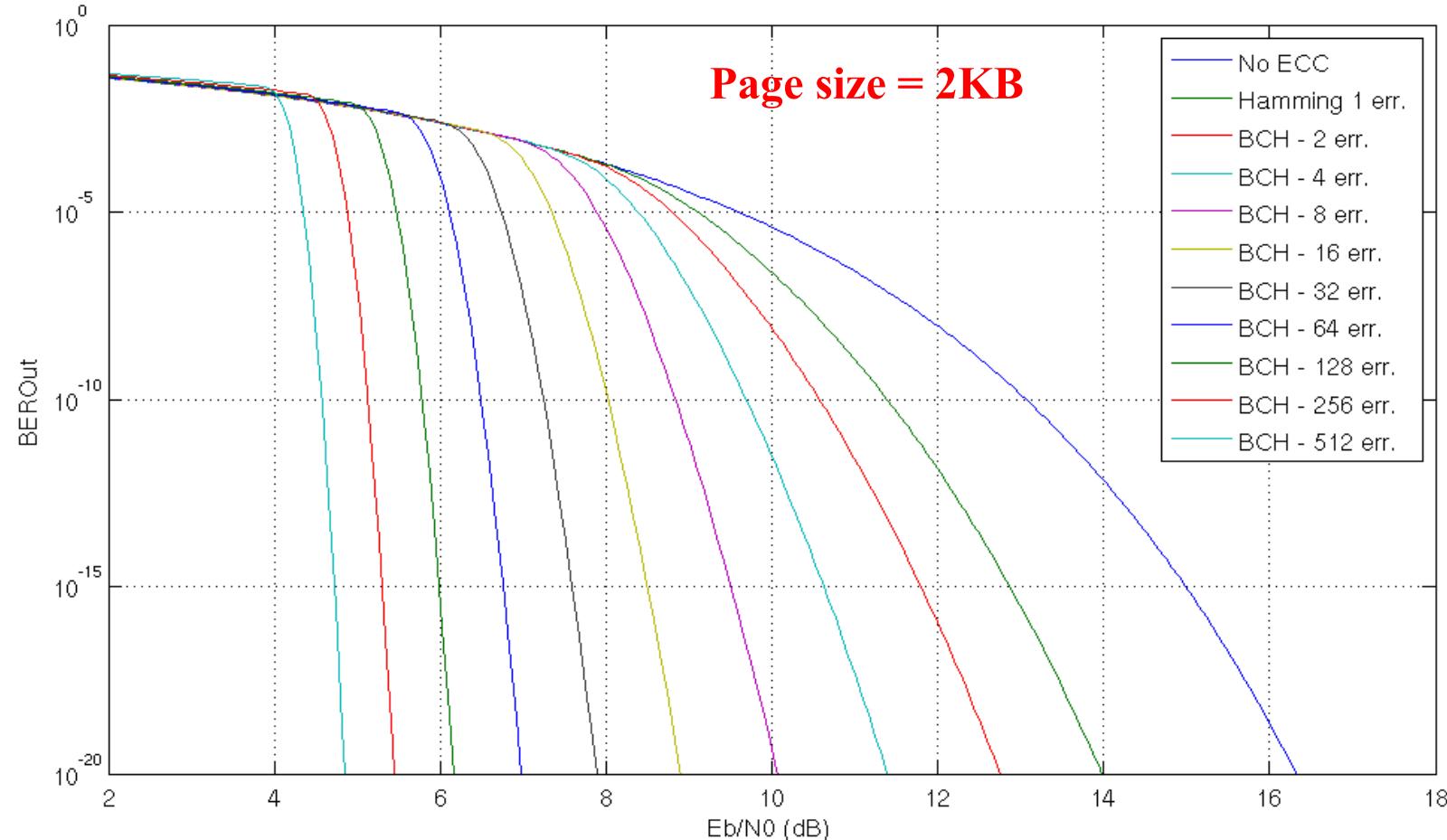


Fig. 4. Dependence of the sector error rate with ECC and cell area overhead on the data length in an ECC word. It is assumed that an error correcting code can correct only one bit in a code word, i.e., single error correcting code (SEC), and the sector error rate without ECC after one million write and erase cycles is 10^{-4} .

ECC in NAND Flash Memory (2/3)

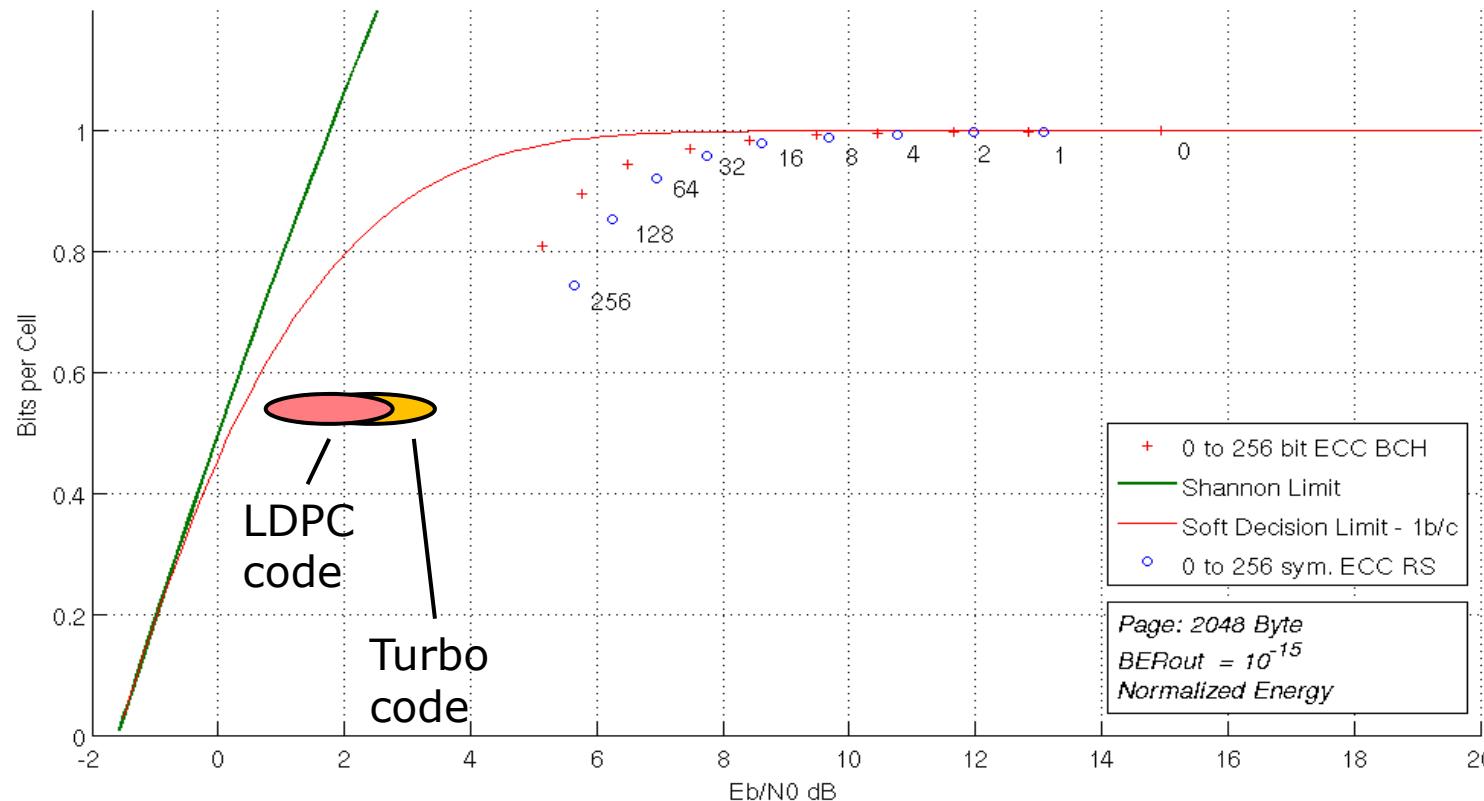
- BCH codes become less effective as code rate decreases



ECC in NAND Flash Memory (2/3)

- BCH codes become less effective as code rate decreases

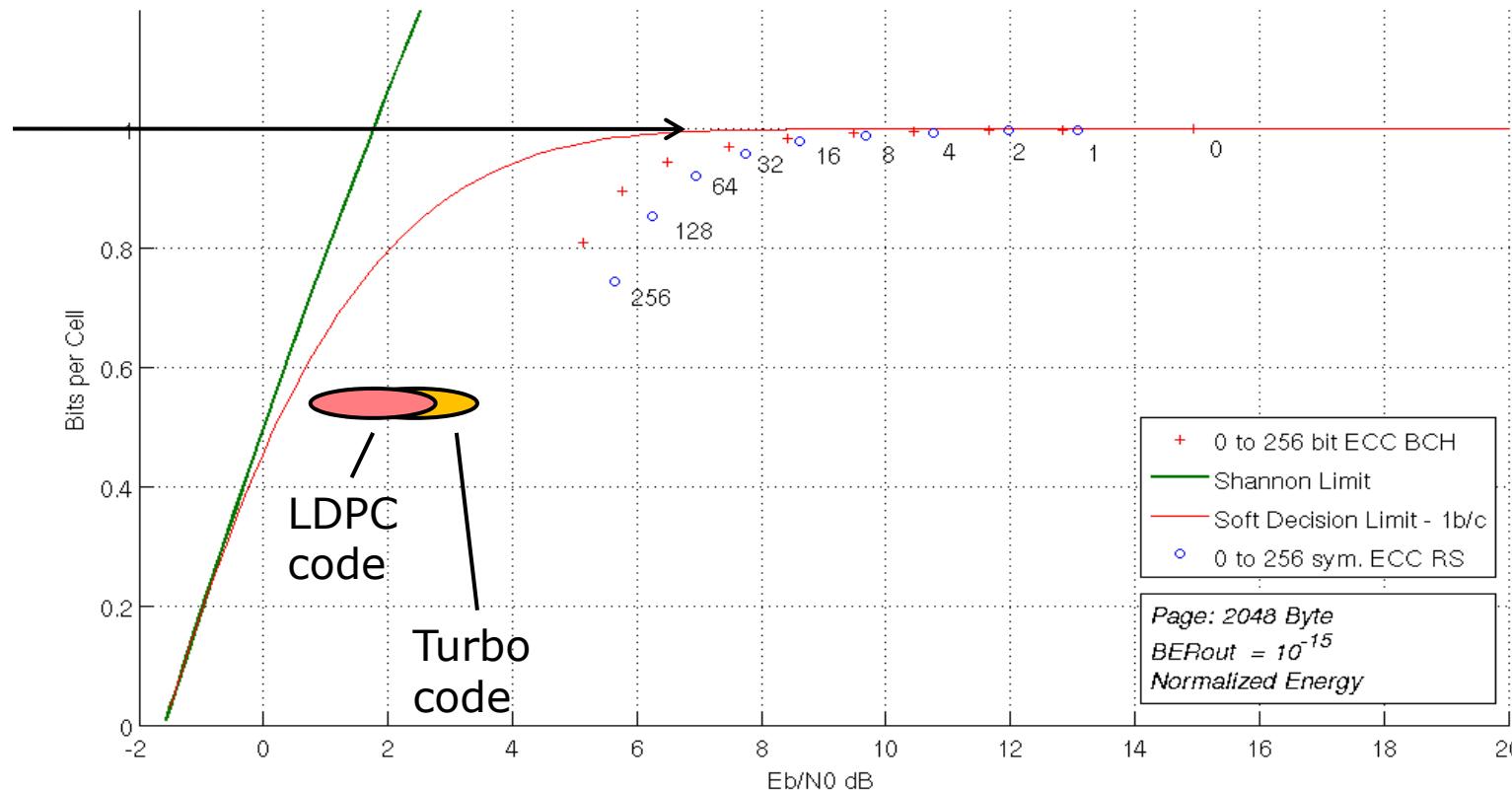
Shannon limit is the theoretical limit of performance



ECC in NAND Flash Memory (2/3)

- BCH codes become less effective as code rate decreases

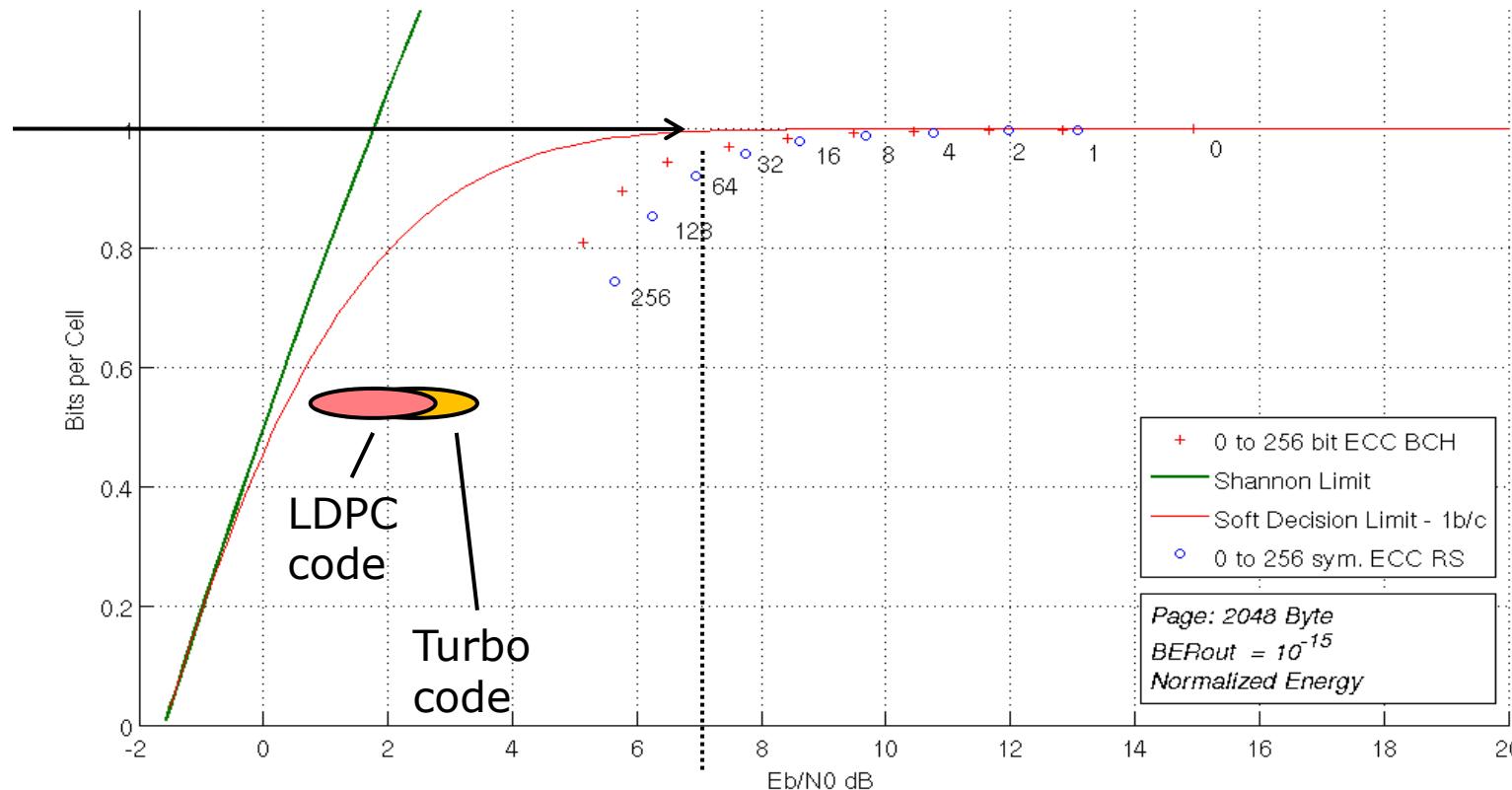
Shannon limit is the theoretical limit of performance



ECC in NAND Flash Memory (2/3)

- BCH codes become less effective as code rate decreases

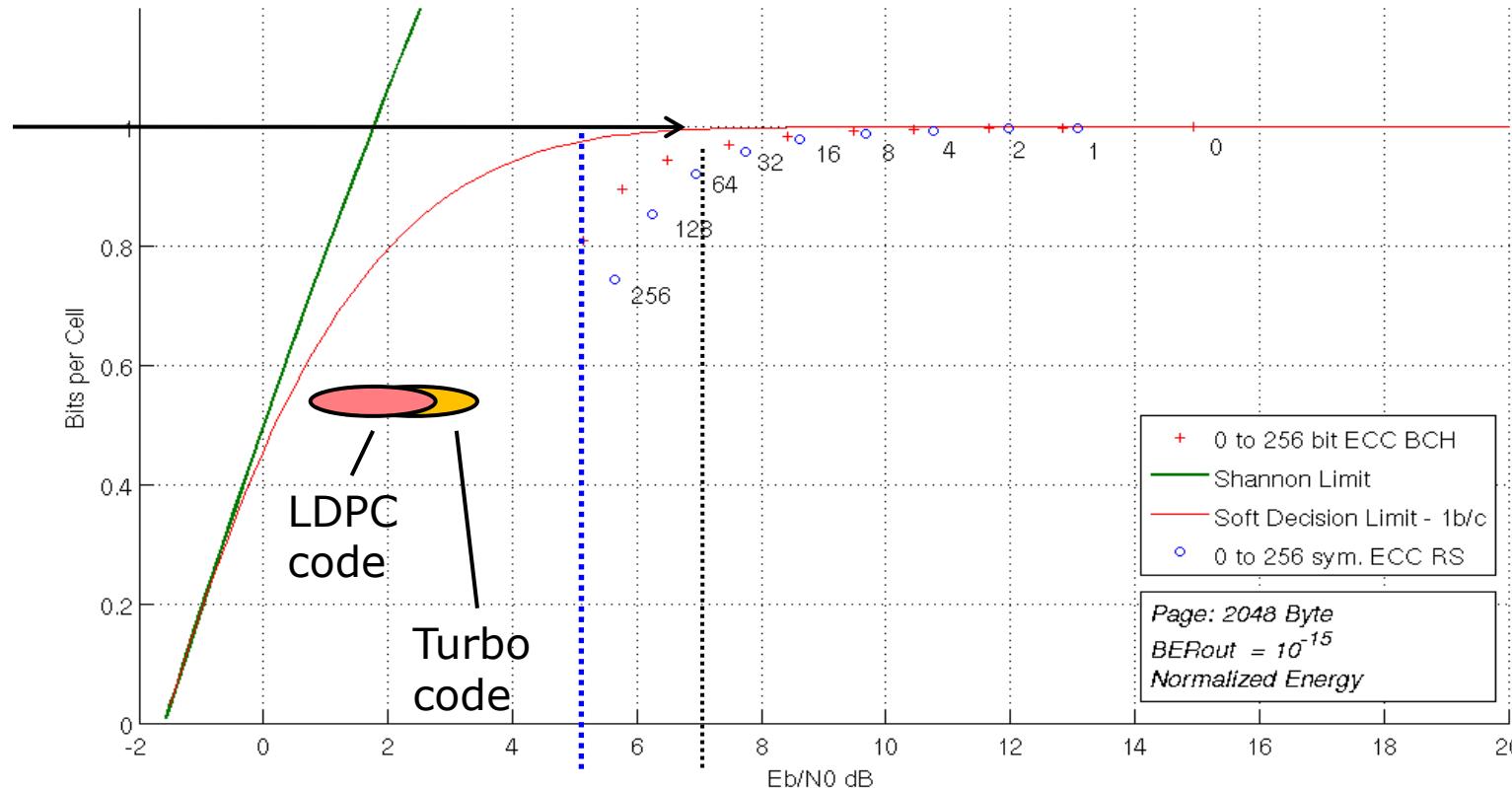
Shannon limit is the theoretical limit of performance



ECC in NAND Flash Memory (2/3)

- BCH codes become less effective as code rate decreases

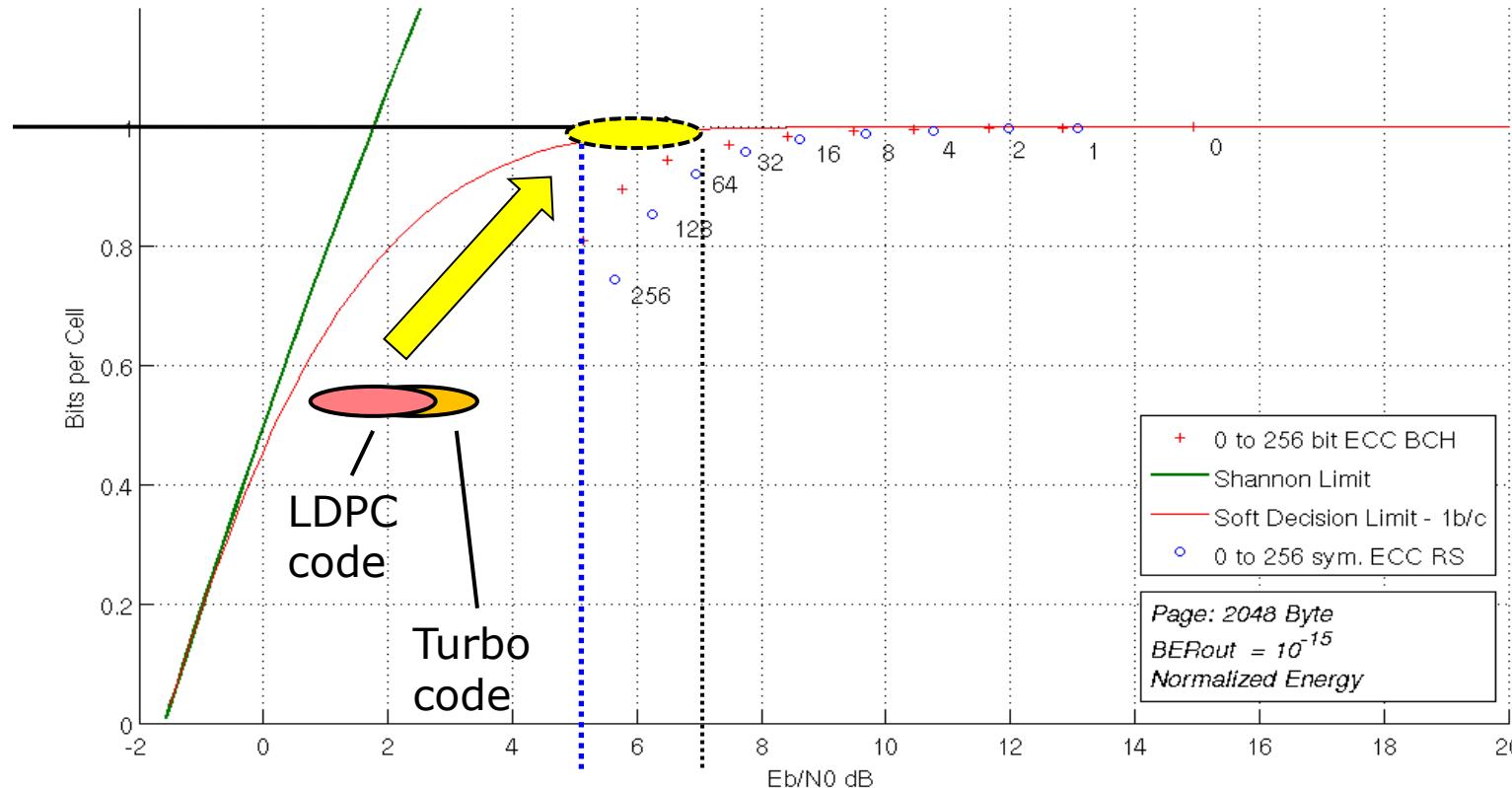
Shannon limit is the theoretical limit of performance



ECC in NAND Flash Memory (2/3)

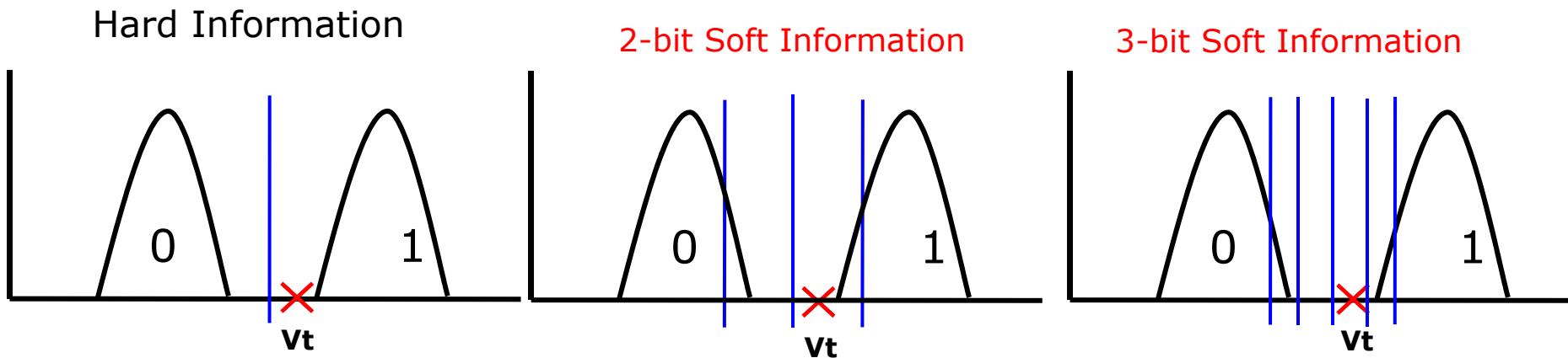
- BCH codes become less effective as code rate decreases

Shannon limit is the theoretical limit of performance



ECC in NAND Flash Memory (3/3)

- Low Density Parity Check (LDPC) codes
 - A linear block code specified by a sparse parity check matrix
 - Presented by Gallager (1963); rediscovered by MacKay (1991)
 - **Probability-based message passing decoding algorithm**
 - Preferred **Soft information** (probability of 1/0)
 - Analog sensing amplifier
 - Increase number of sensing → slower read



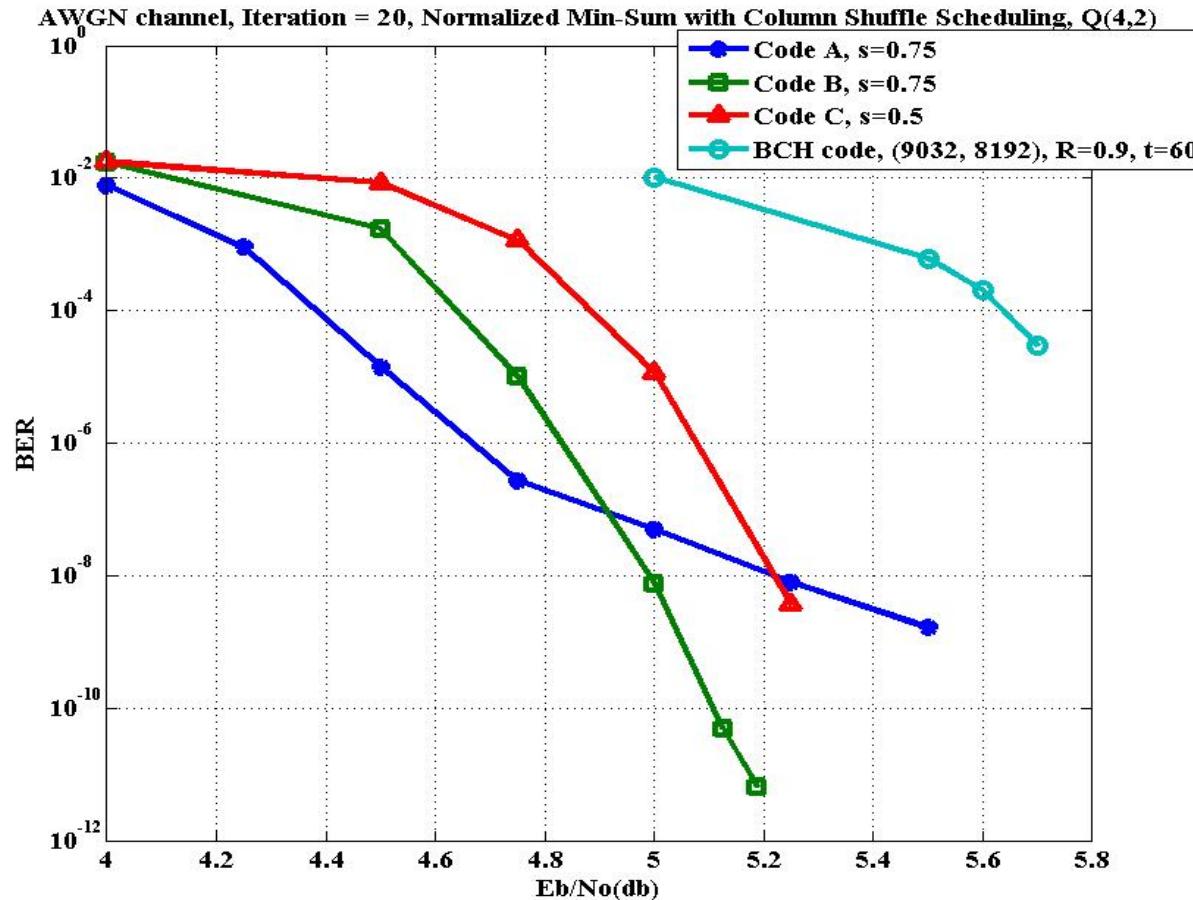
BCH Codes vs. LDPC Codes (1/3)

- BCH code
 - Deterministic decoding algorithm (algebraic)
- LDPC code
 - Non-deterministic decoding algorithm (probability-based)
- Example

Error bits	110	130	150	170	190
BCH (t=150)	100% OK	100% OK	100% OK	100% fail	100% fail
LDPC	99.9999% OK	99.999% OK	99.99% OK	95% OK	90% OK

BCH Codes vs. LDPC Codes (2/3)

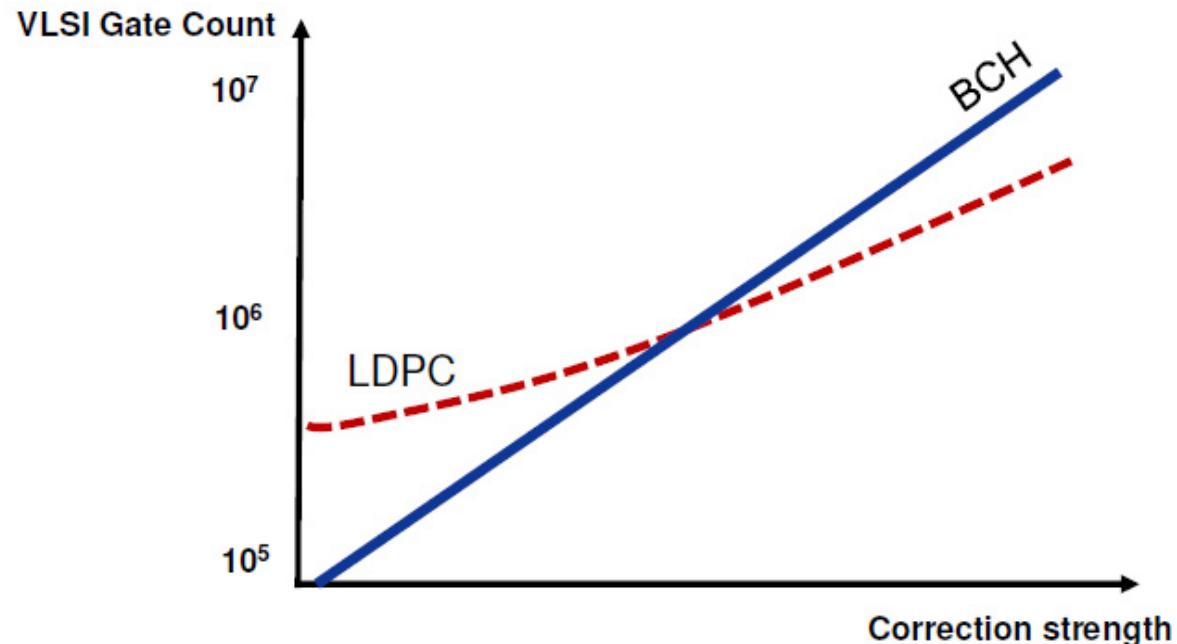
□ Performance Analysis



Code A: (9366, 8474) with dv=4
Code B: (9387, 8493) with dv=6
Code C: (9153, 8249) with dv=8

BCH Codes vs. LDPC Codes (3/3)

□ Complexity Comparison



- LDPC decoders have a higher initial barrier, but complexity does not increase as quickly as BCH decoders at higher correction strengths.

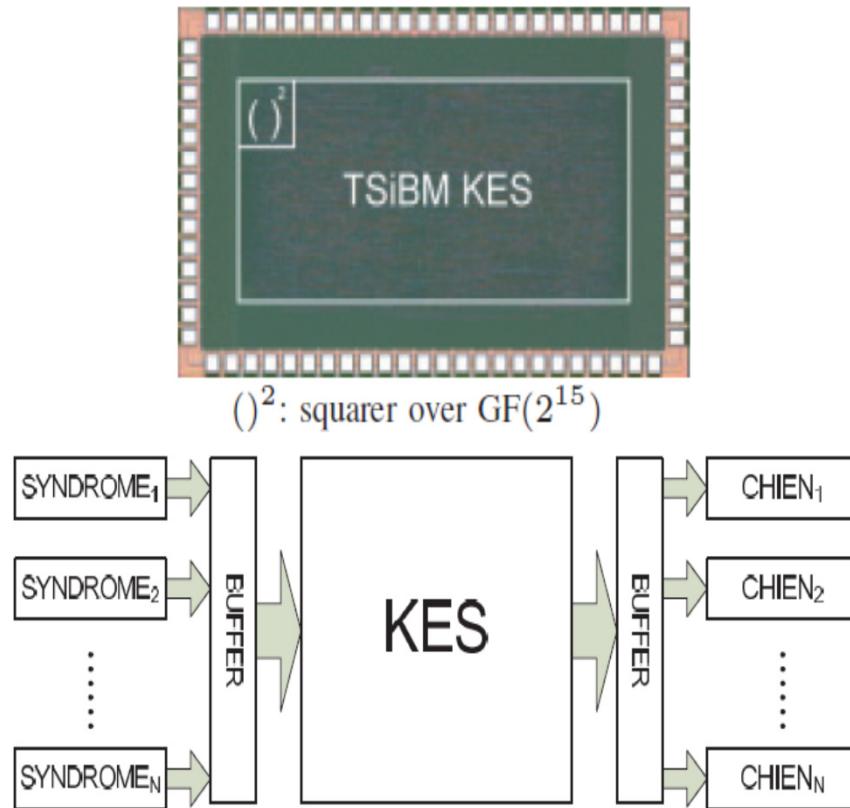
Flash Memory Summit 2012
Santa Clara, CA



13

Multi-Channel BCH Decoder Architecture

- An Efficient BCH Decoder with 124-bit Correctability for Multi-Channel SSD Applications," presented in IEEE ASSCC 2012.



	Design I	Design II	K.Lee [9]	Y.Lee [10]
Technology	90nm	90nm	45nm	$0.13\mu m$
Data size	2KB	2KB	1KB	1KB
Correction capability	124bits	124bits	64bits	32bits
Parallel factor	8	16	16	32
Support channel #	8	8	1	4
Simultaneously Support channel #	8	8	1	1
Clock rate	400MHz	400MHz	400MHz	200MHz
Throughput	25.6Gb/s	51.2Gb/s	6.4Gb/s	6.4Gb/s
Total Gate-Count	1315.6K*	2110.3K**	230K	110K
Avg. Gate-Count	164.5K	263.8K	230K	110K

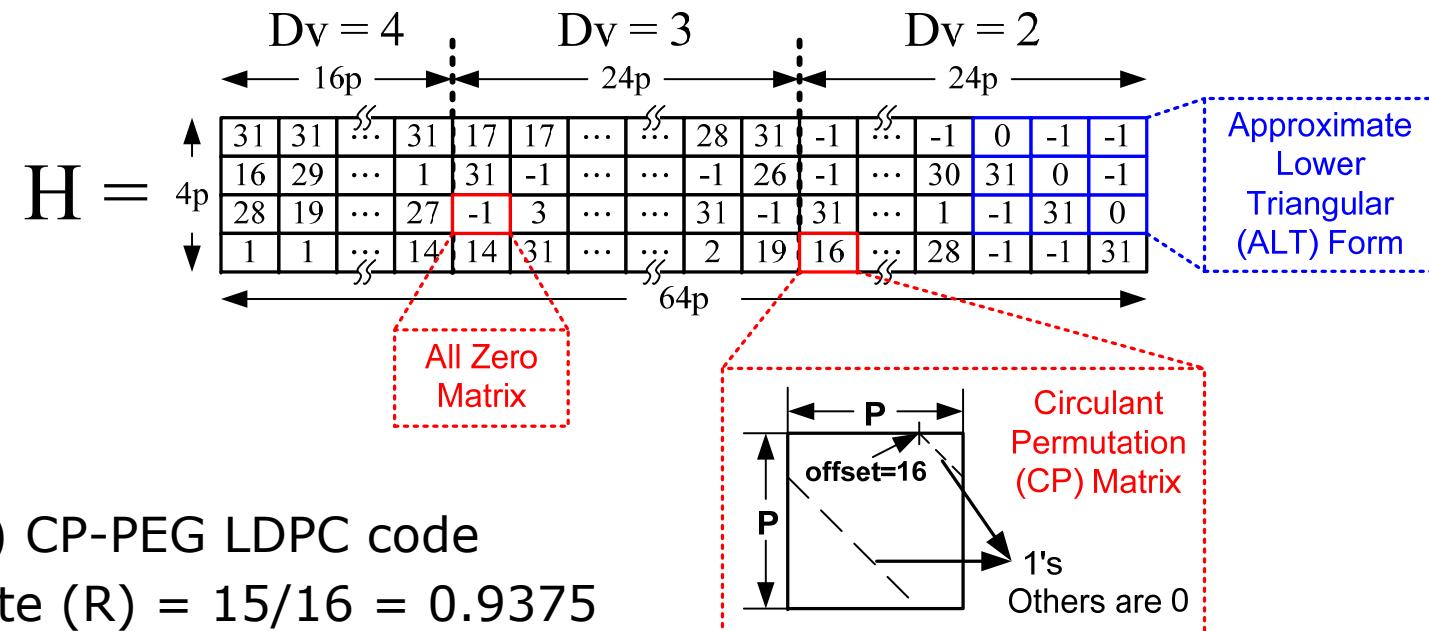
[*] sum up with gate count $SG_{half}=38.4K$, KES with squarer=250.0K and CS=94.8K

[**] sum up with gate count SG=63.6K, KES=239.1K and CS=170.3K

CP-PEG LDPC Decoder Architecture (1/3)

□ Circulant Permutation Progressive Edge-Growth Algorithm

- Better performance than other PEG-based LDPC codes
- Lower encoding complexity by ALT encoding



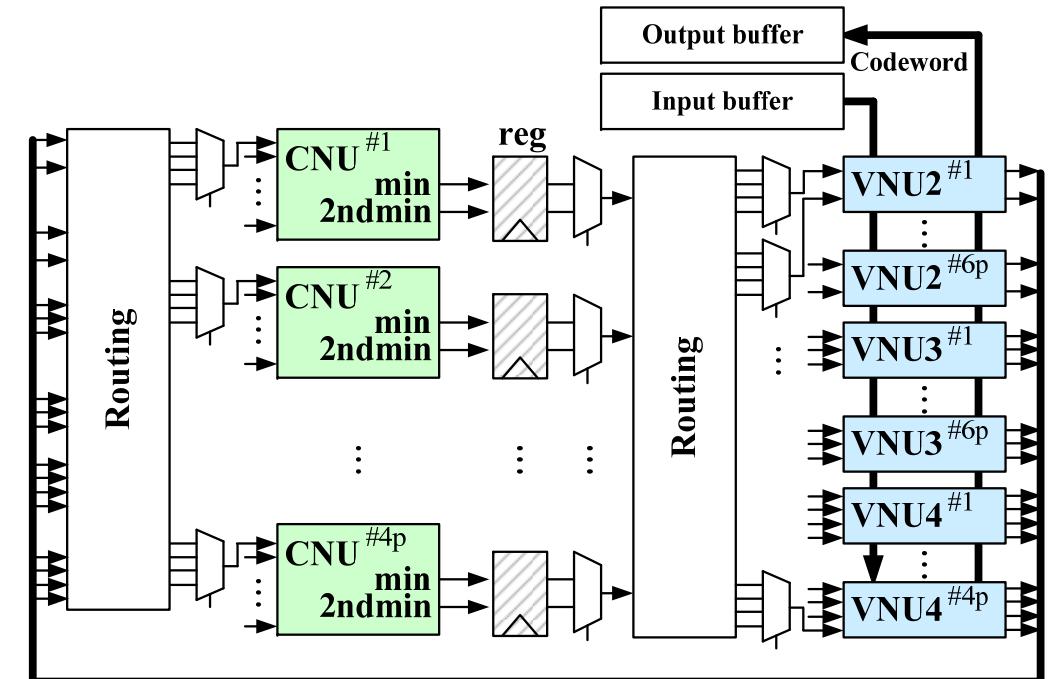
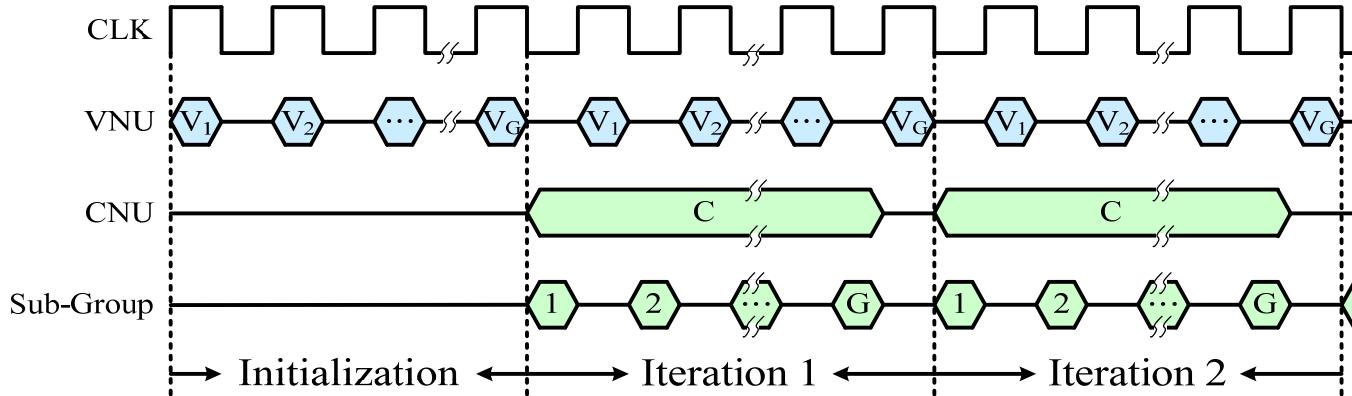
□ Example

- (2048, 1920) CP-PEG LDPC code
- High code-rate (R) = $15/16 = 0.9375$
- Large check-node degree (Dc) = 45
- Cyclically shifted identity sub-matrix with size (p) = 32

CP-PEG LDPC Decoder Architecture (2/3)

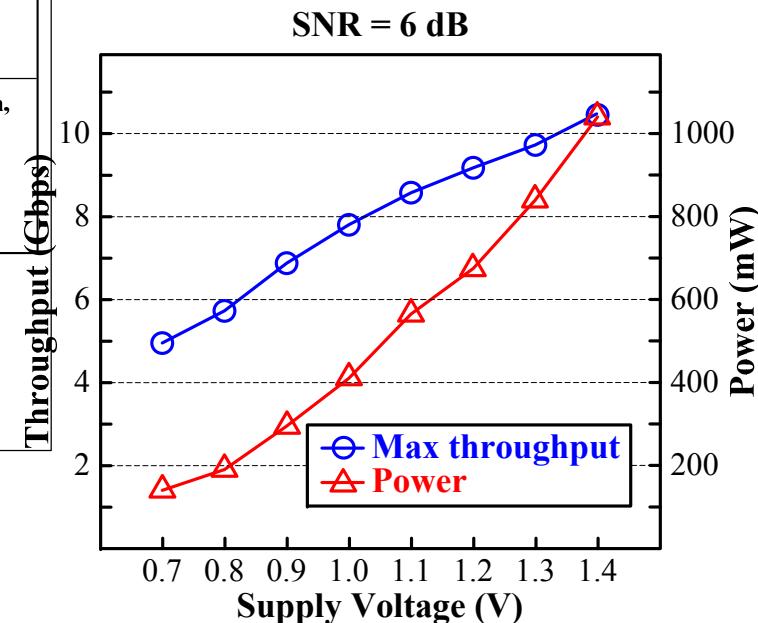
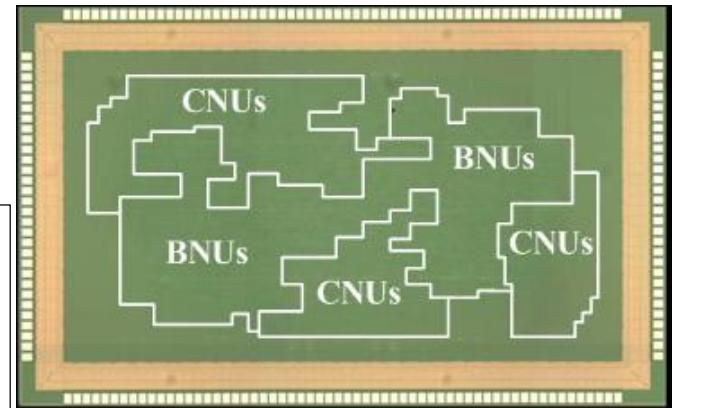
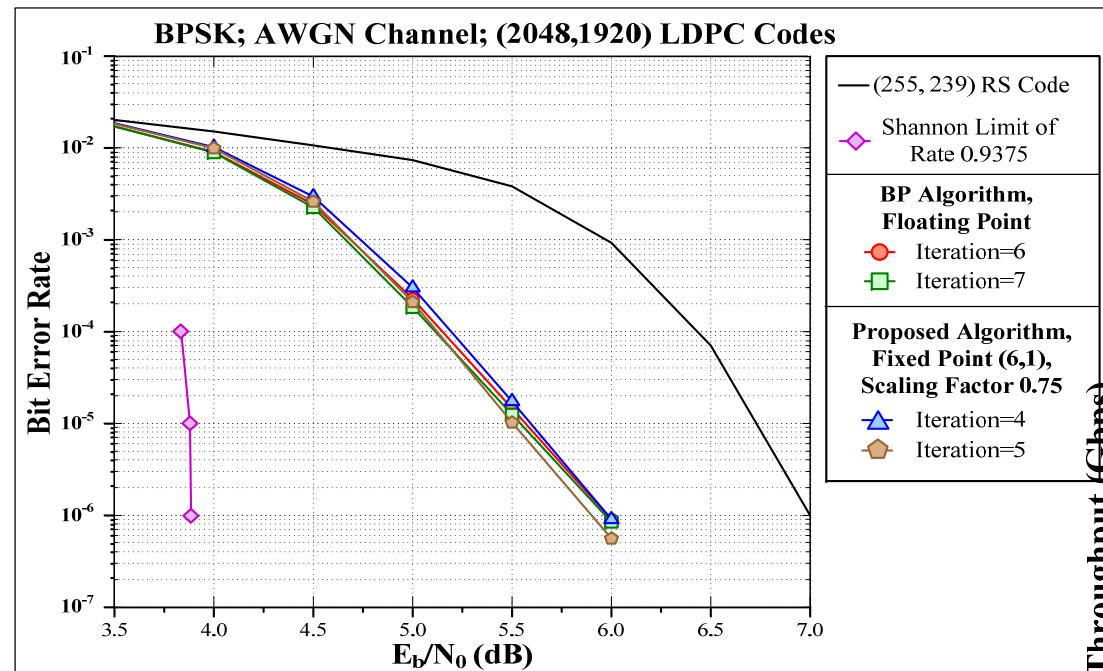
□ Single Pipelined Architecture

- Complete CNU and VNU operations in one cycle



CP-PEG LDPC Decoder Architecture (3/3)

Chip Measurement Result

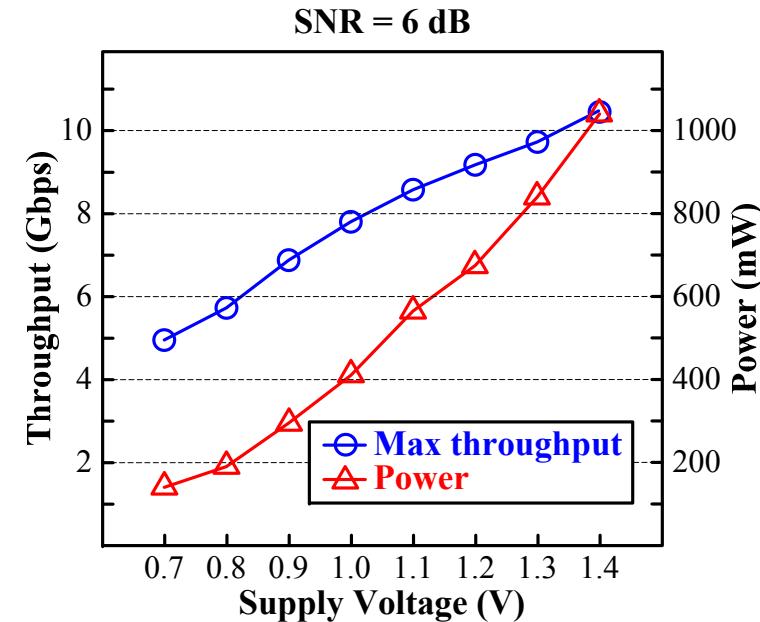
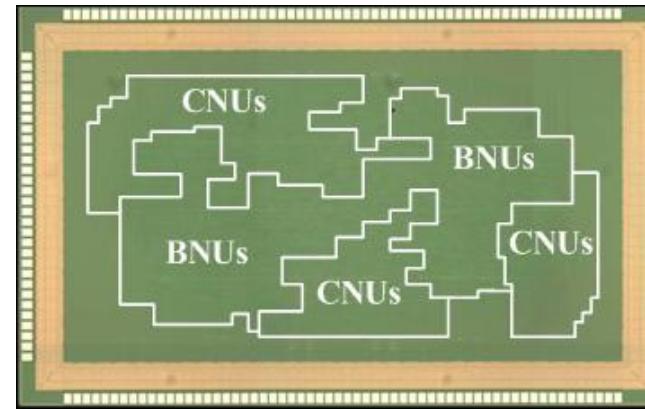


CP-PEG LDPC Decoder Architecture (3/3)

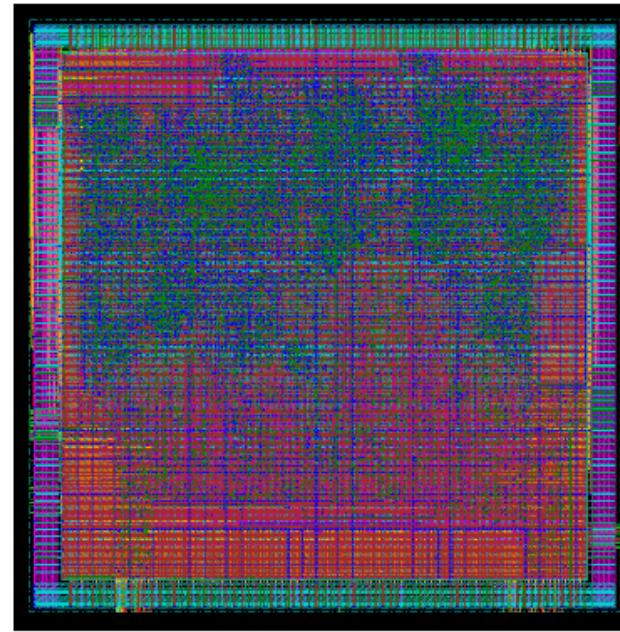
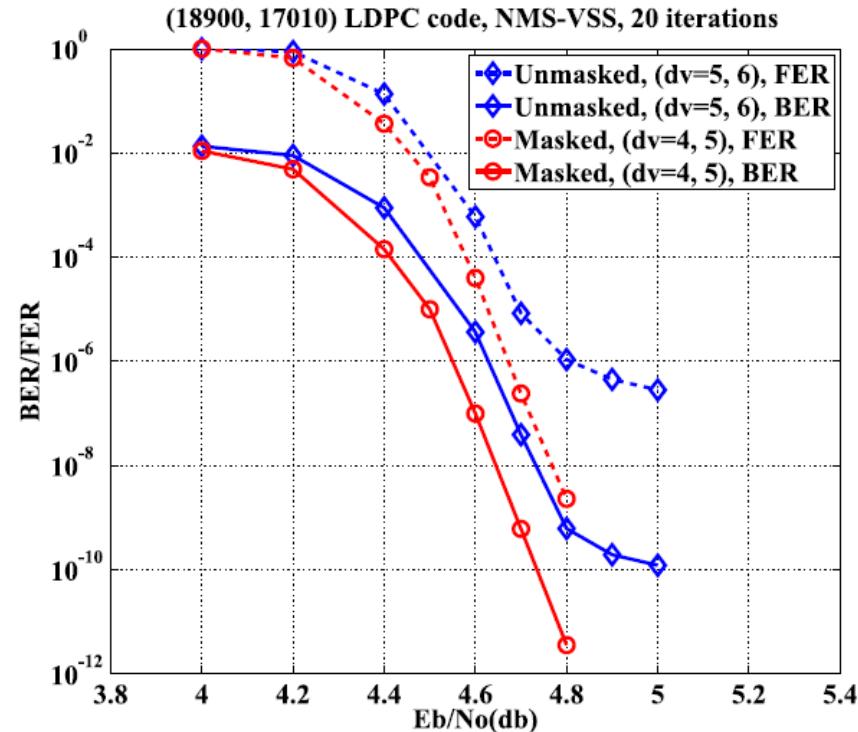
□ Chip Measurement Result

Process	UMC 90nm 1P9M
Code Spec	(2048, 1920)
Input Quantization	6 bits
Iteration	4
Clock Frequency	120 MHz
Max. Throughput	11.5 Gbps
Gate Count	708 k
Core Area	3.84 mm ²
Utilization	68%
Power	191 mW (@0.8 V)
Energy Efficiency	0.033 nJ/bit (@0.8V)

Note : Measurement result of FF corner chip



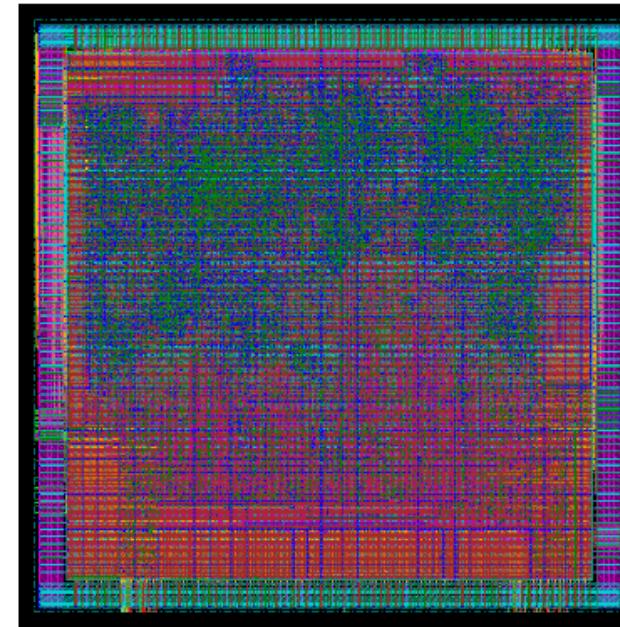
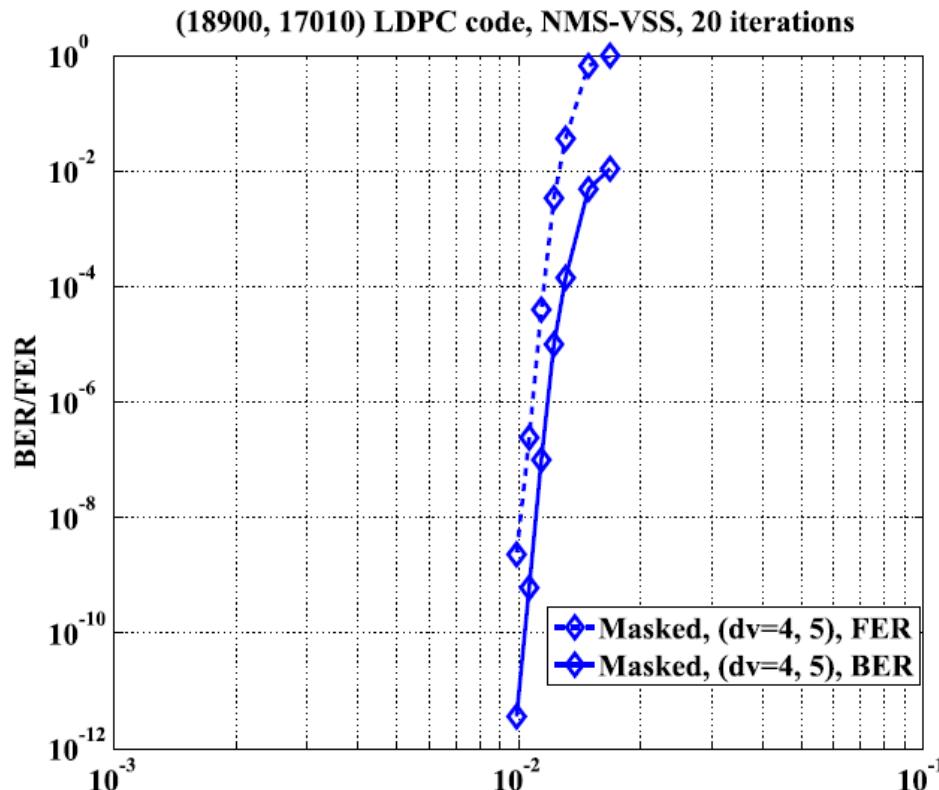
QC-LDPC LDPC Decoder Architecture



	Gate Count
VNU	110K
CNU & min selector	160K
Memory	Min & Index
	Global sign bit
	Channel value
	Sign bit
	Hard decision
	Registers & control circuits
Total	520K

- A 520K (18900,17010) Array Dispersion LDPC Decoder Architecture for NAND Flash Memory," IEEE T-VSLI 2016.

QC-LDPC LDPC Decoder Architecture

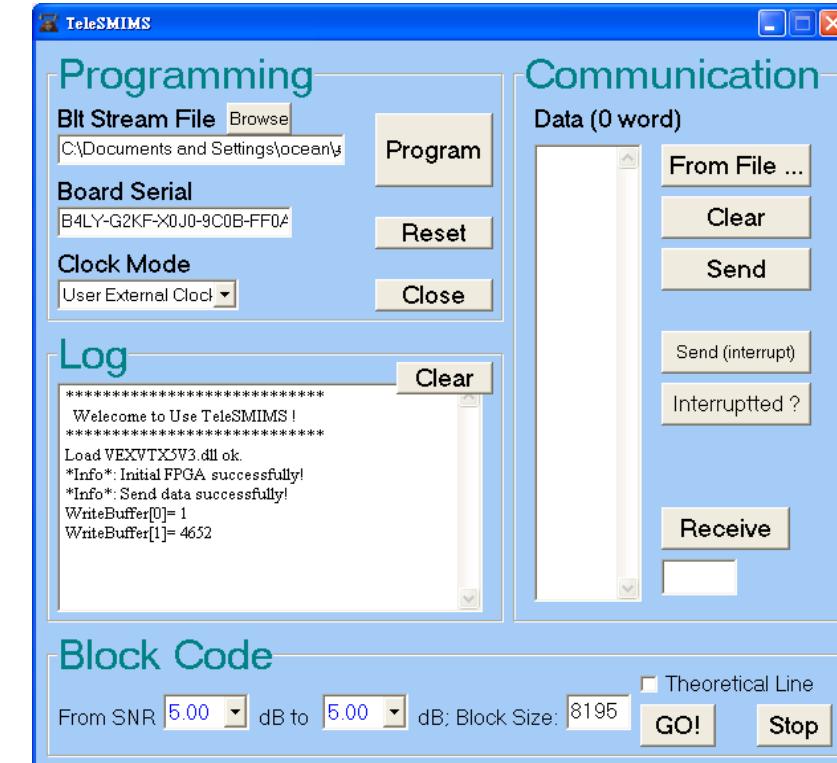
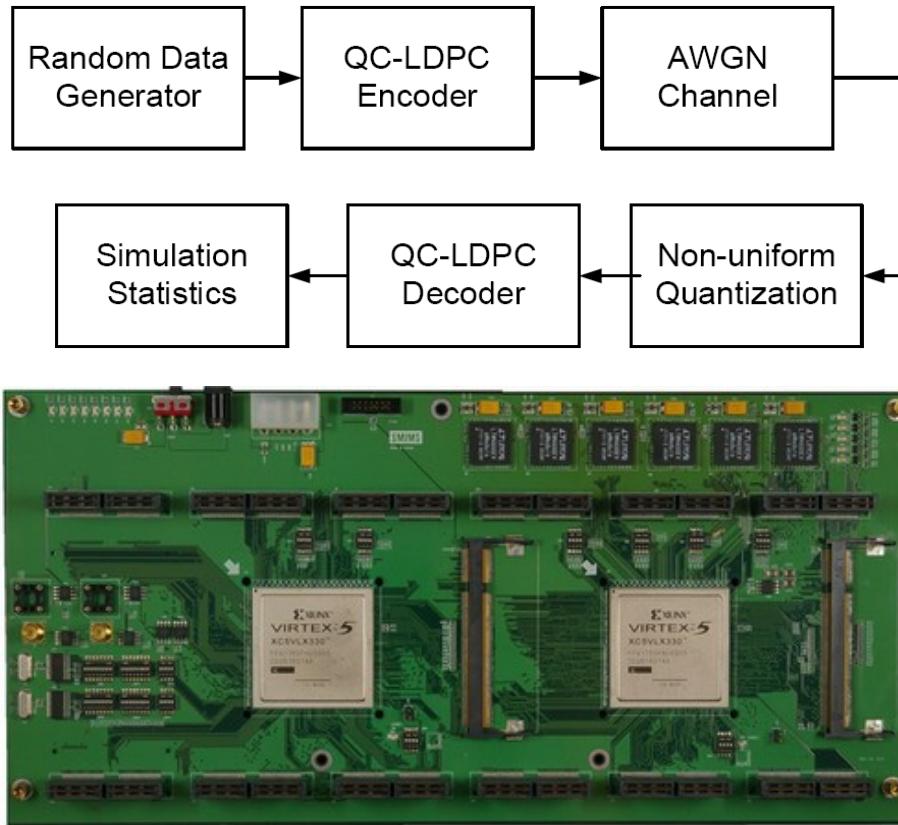


	Gate Count
VNU	110K
CNU & min selector	160K
Min & Index	74K
Global sign bit	9K
Channel value	25K
Sign bit	74K
Hard decision	14K
Registers & control circuits	54K
Total	520K

- A 520K (18900,17010) Array Dispersion LDPC Decoder Architecture for NAND Flash Memory," IEEE T-VSLI 2016.

Error Floor Investigating Platform

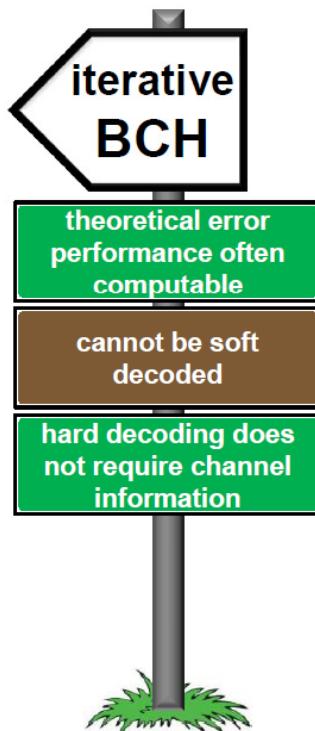
- The QC-LDPC codes can be evaluated down to BER of 10^{-12} within one day



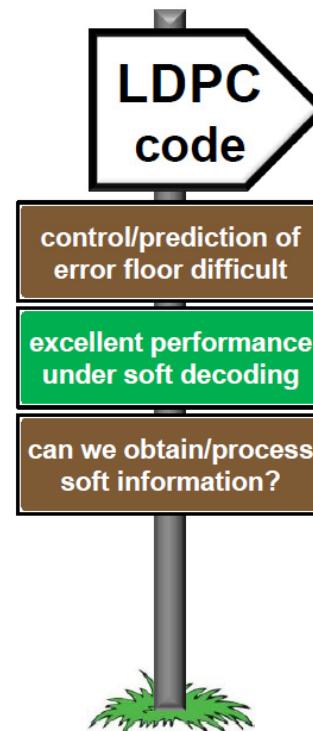
What is NEXT?



ECC for SSD: Where We Go Next



Flash Memory Summit 2015
Santa Clara, CA



4

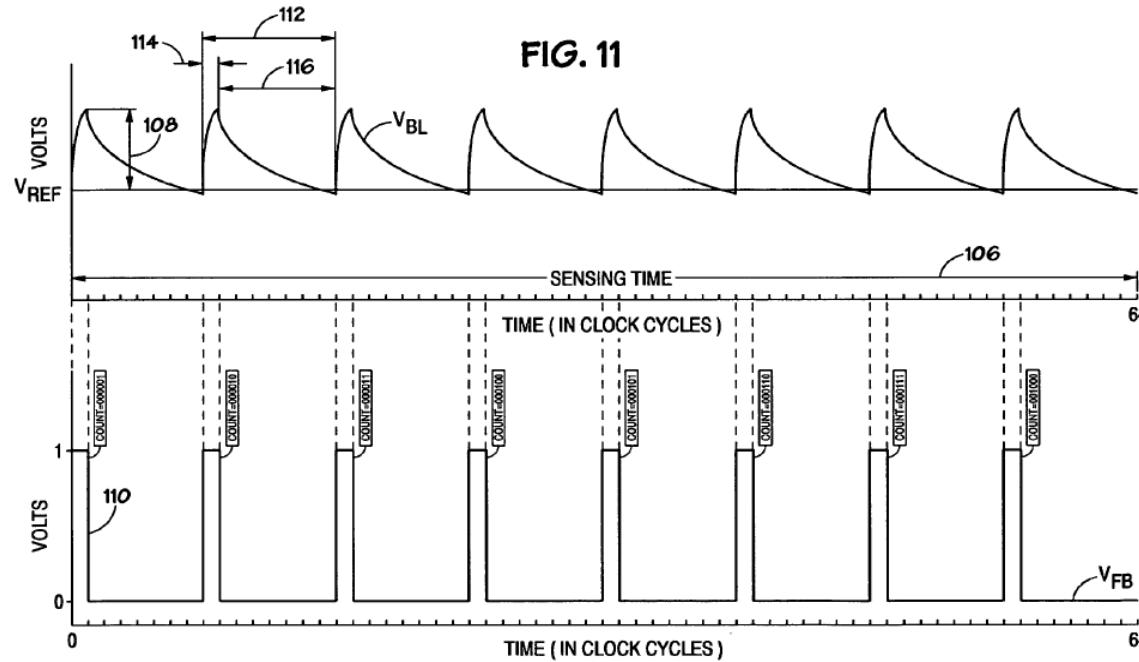
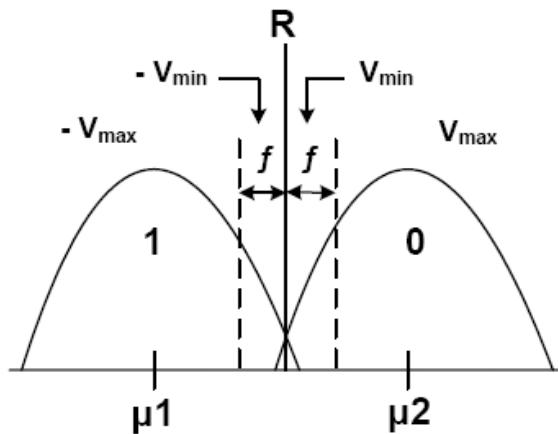
Channel model (1/2)

□ Hard information

- Suitable for algebraic decoding algorithm such as BCH/RS codes

□ Soft information

- Requires multi-sensing or analog sensing

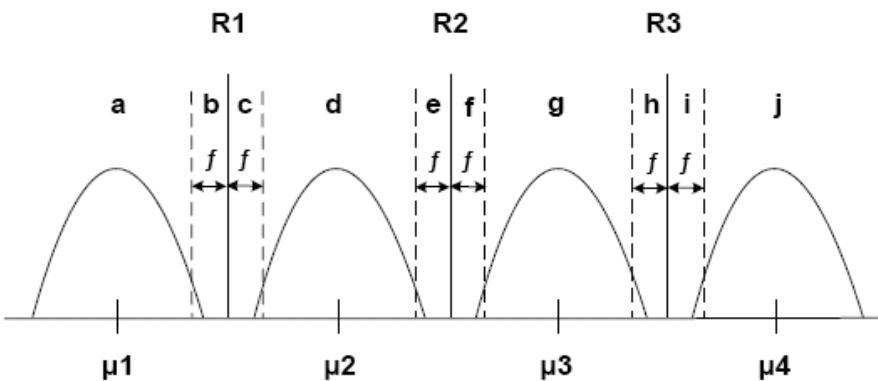


Channel model (2/2)

□ Soft information (cont'd)

■ Multi-sensing scheme

	Number of sensing	
	High page data	Low page data
Hard information	1	2
2-bit Soft information	3	6



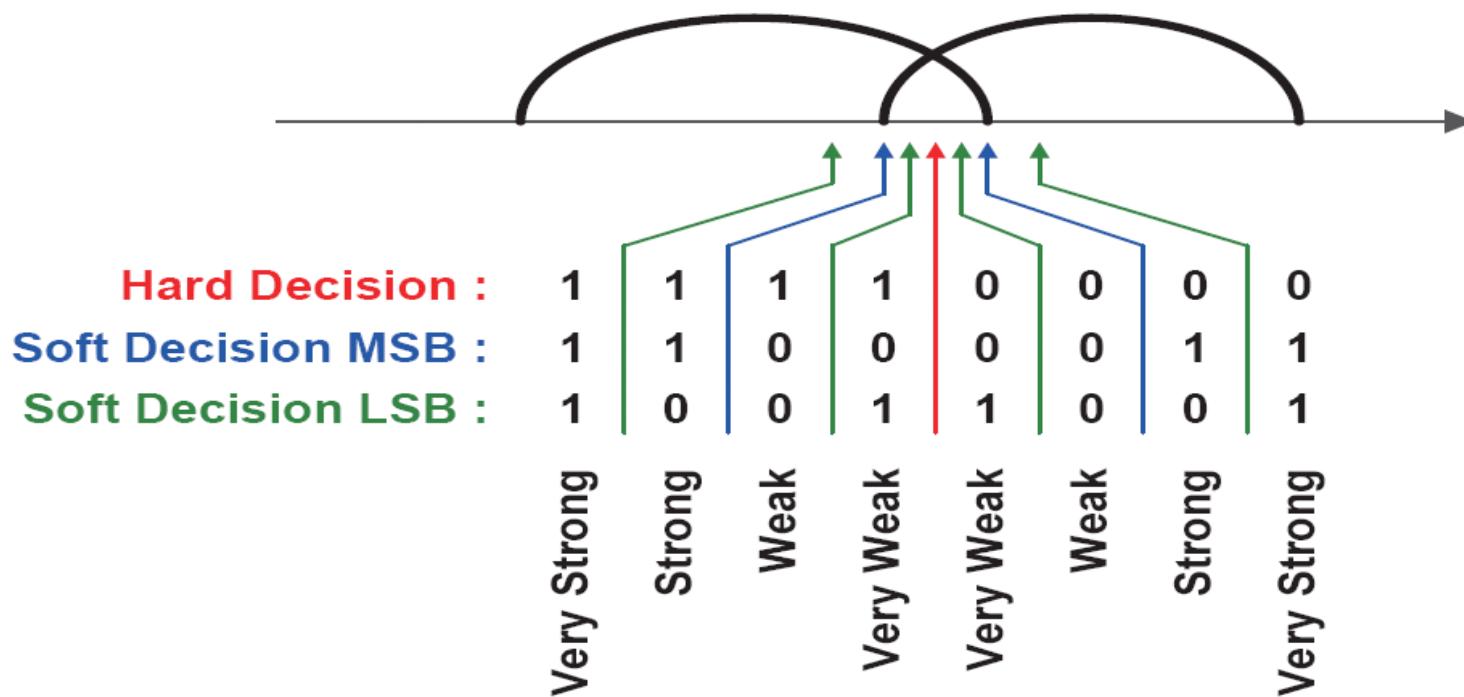
Non-uniform quantization

Regions	1 st data bit	2 nd data bit
a: $V_t \leq R_1 - f$	- V_{max}	- V_{max}
b: $R_1 - f < V_t \leq R_1$	- V_{max}	- V_{min}
c: $R_1 < V_t \leq R_1 + f$	- V_{max}	+ V_{min}
d: $R_1 + f < V_t \leq R_2 - f$	- V_{max}	+ V_{max}
e: $R_2 - f < V_t \leq R_2$	- V_{min}	+ V_{max}
f: $R_2 < V_t \leq R_2 + f$	+ V_{min}	+ V_{max}
g: $R_2 + f < V_t \leq R_3 - f$	+ V_{max}	+ V_{max}
h: $R_3 - f < V_t \leq R_3$	+ V_{max}	+ V_{min}
i: $R_3 < V_t \leq R_3 + f$	+ V_{max}	- V_{min}
j: $R_3 + f < V_t$	+ V_{max}	- V_{max}

Soft information in NAND Flash (1/3)

□ Samsung's approach in VLSI Symposium 2011

- Chulbum Kim et. al., "A 21nm High Performance 64Gb MLC NAND Flash Memory with 400MB/s Asynchronous Toggle DDR Interface,"



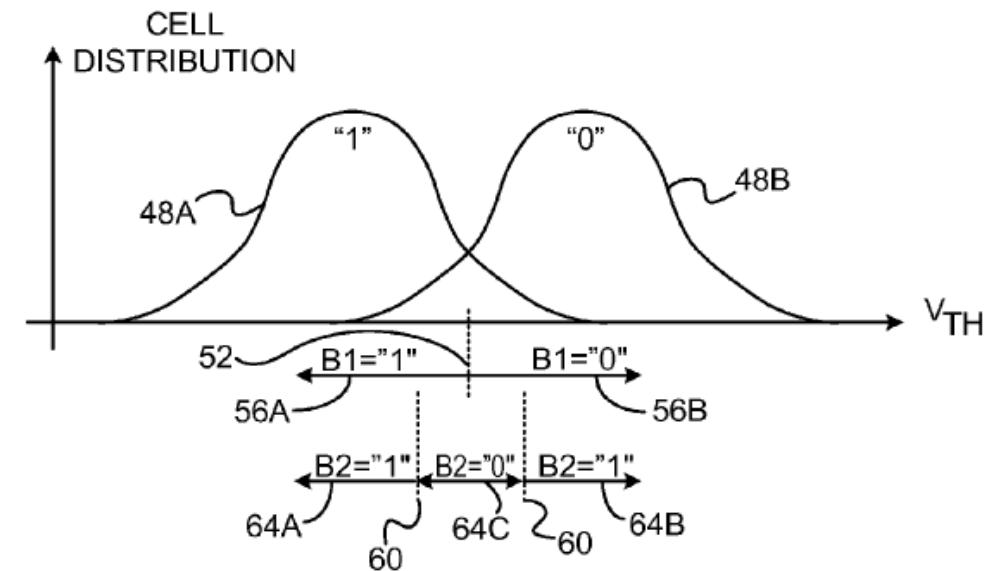
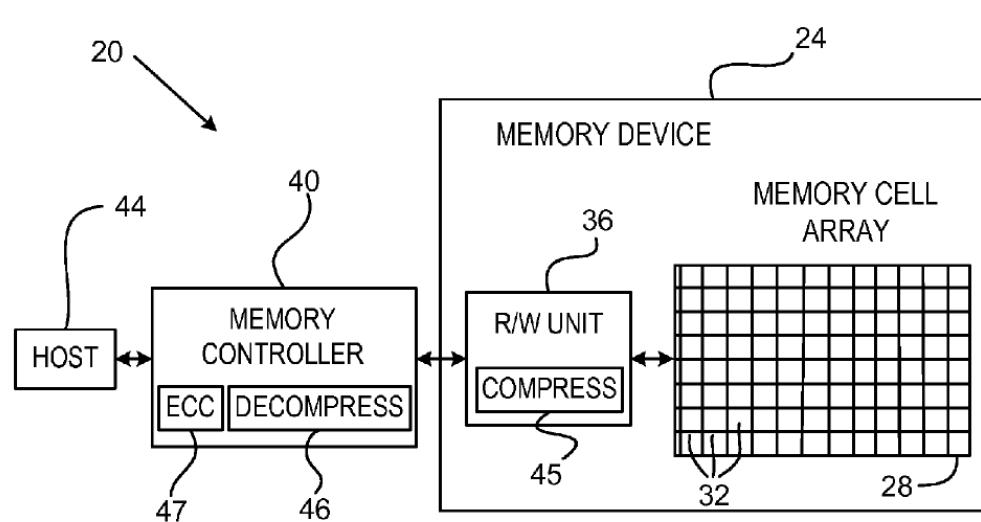
Soft information in NAND Flash (2/3)

□ Anobit's Patent

- Efficient Readout From Analog Memory Cells using Data Compression

(12) **United States Patent**
Perlmutter et al.

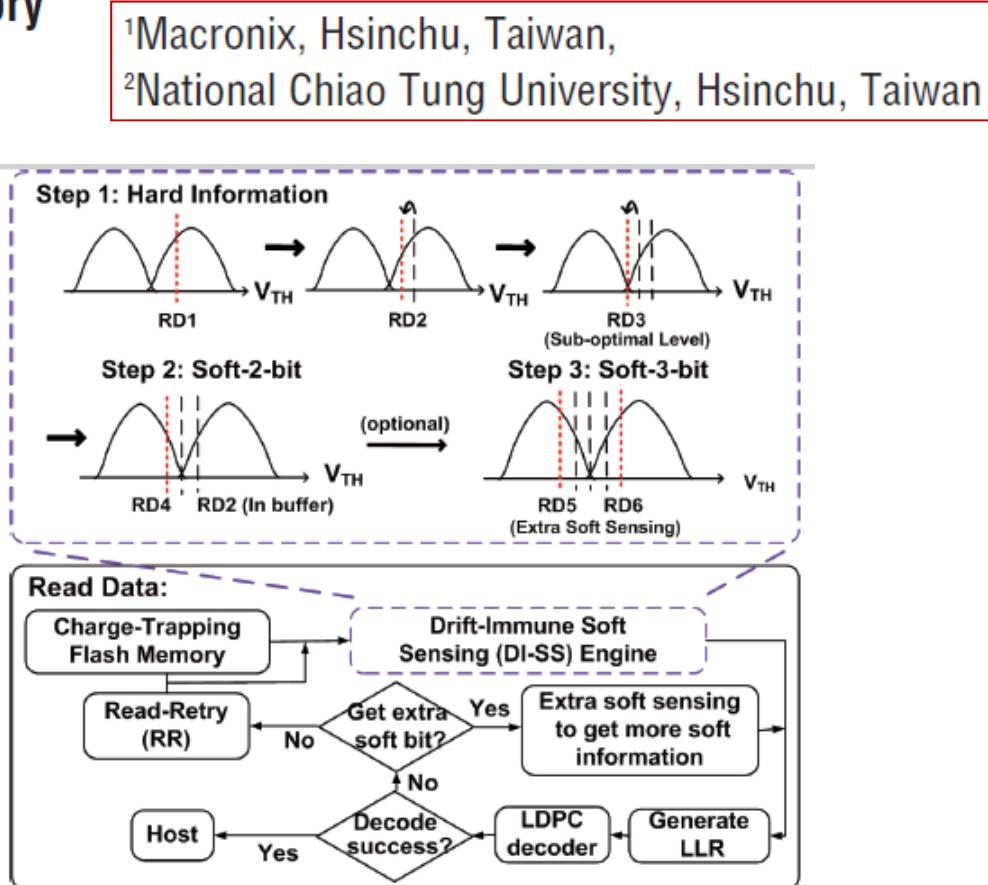
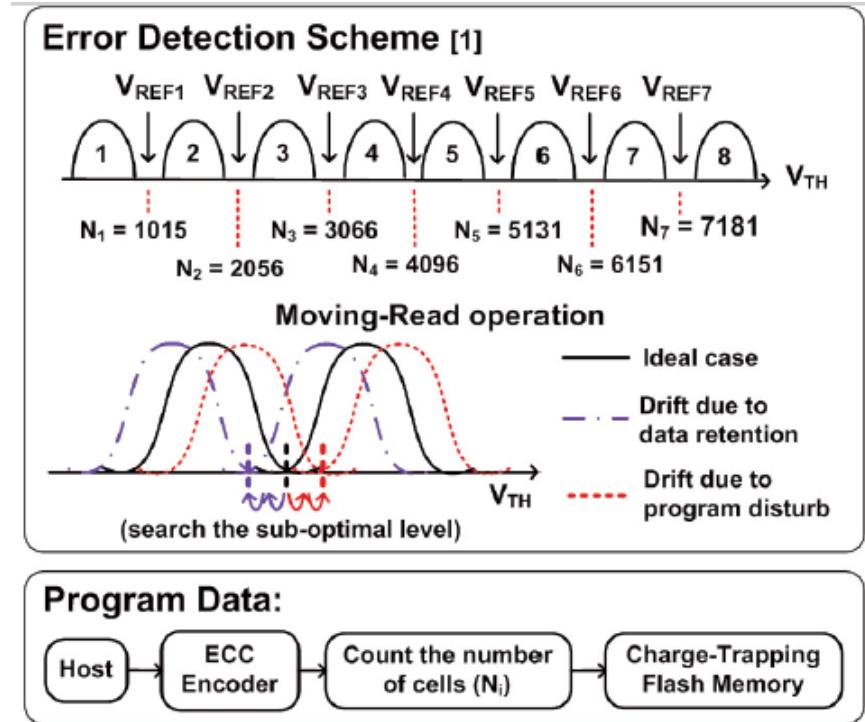
(10) **Patent No.:** US 8,230,300 B2
(45) **Date of Patent:** Jul. 24, 2012



Soft information in NAND Flash (3/3)

□ In ISSCC 2013,

12.7 A 45nm 6b/cell Charge-Trapping Flash Memory Using LDPC-Based ECC and Drift-Immune Soft-Sensing Engine



Summary

- In this talk, we cover ECC technologies for 5G/NVM applications

- Both aiming for higher throughput and lower error rate
- Different design philosophy and trend can be listed as

	ECC for 5G communication	ECC for NVM systems
Block length	Variable lengths and covers a wide range, from hundreds to thousand	Usually fixed and large 2K/4K/8K bytes or more
Code rate	Very wide range (ex. 1/3 ~ 8/9)	Usually higher than 0.85
Channel condition	Large variation	Degrades (gradually) with time
Error correction capability	BER < 10^{-5} , good in waterfall region	BER < 10^{-12} , low error floor
Candidates	LDPC and Polar codes	BCH and LDPC

- To achieve systematic approach, it is a MUST to derive hardware architecture based on some specific priority (power/cost/performance)

Reference

- Junae Li, Shu Lin, Khaled Abdel-Ghaffar, W.E. Ryan, and D.J. Costello, ***LDPC Code Designs, Constructions, and Unification***, Cambridge, 2016
- C.-C Wong and H.-C Chang, ***Turbo Decoder Architecture for Beyond-4G Applications***, Springer, 2013.
- W.E. Ryan and Shu Lin, ***Channel codes – Classical and Modem***, Cambridge, 2010
- R.H. Morelos-Zaragoza, ***The Art of Error Correcting Coding***, 2nd ed., Wiley, 2006
- Todd K. Moon, ***Error Correction Coding***, Wiley, 2005
- Shu Lin and D.J. Costello, ***Error-Control Coding***, 2nd ed., Prentice Hall, 2004
- R.J. McEliece, ***The Theory of Information and Coding***, 2nd ed., Cambridge, 2002.

Reference

- Junae Li, Shu Lin, Khaled Abdel-Ghaffar, W.E. Ryan, and D.J. Costello, ***LDPC Code Designs, Constructions, and Unification***, Cambridge, 2016
- C.-C Wong and H.-C Chang, ***Turbo Decoder Architecture for Beyond-4G Applications***, Springer, 2013.
- W.E. Ryan and Shu Lin, ***Channel codes – Classical and Modem***, Cambridge, 2010
- R.H. Morelos-Zaragoza, ***The Art of Error Correcting Coding***, 2nd ed., Wiley, 2006
- Todd K. Moon, ***Error Correction Coding***, Wiley, 2005
- Shu Lin and D.J. Costello, ***Error-Control Coding***, 2nd ed., Prentice Hall, 2004
- R.J. McEliece, ***The Theory of Information and Coding***, 2nd ed., Cambridge, 2002.

- “Brief Introduction to Turbo-Like Codes,” ***Ocean group***, NCTU, 2013.

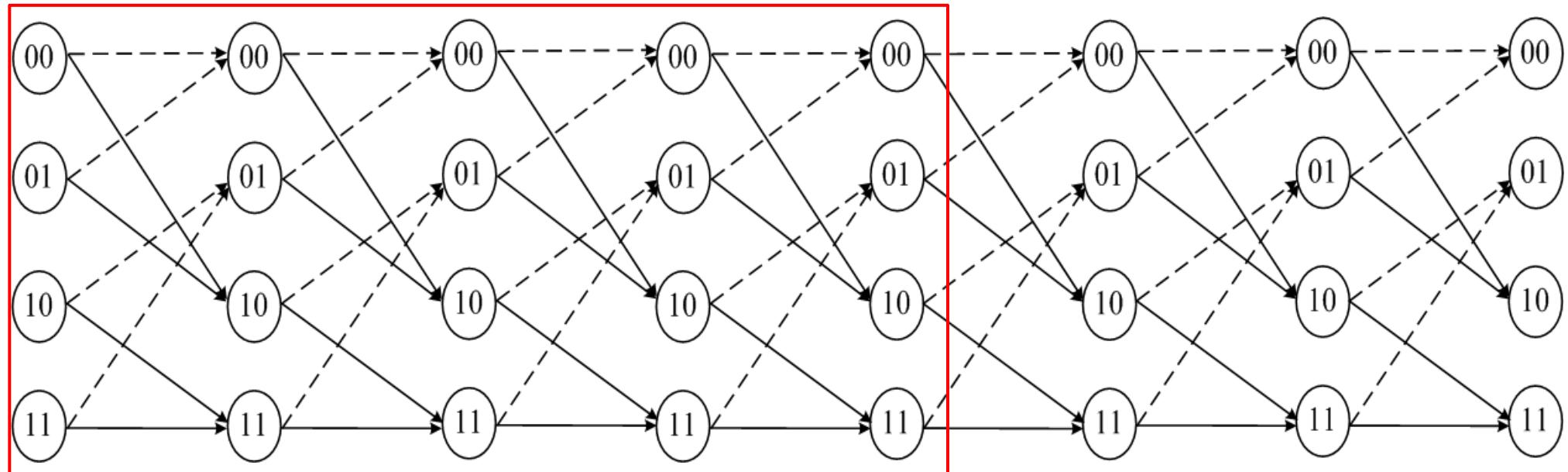
Reference

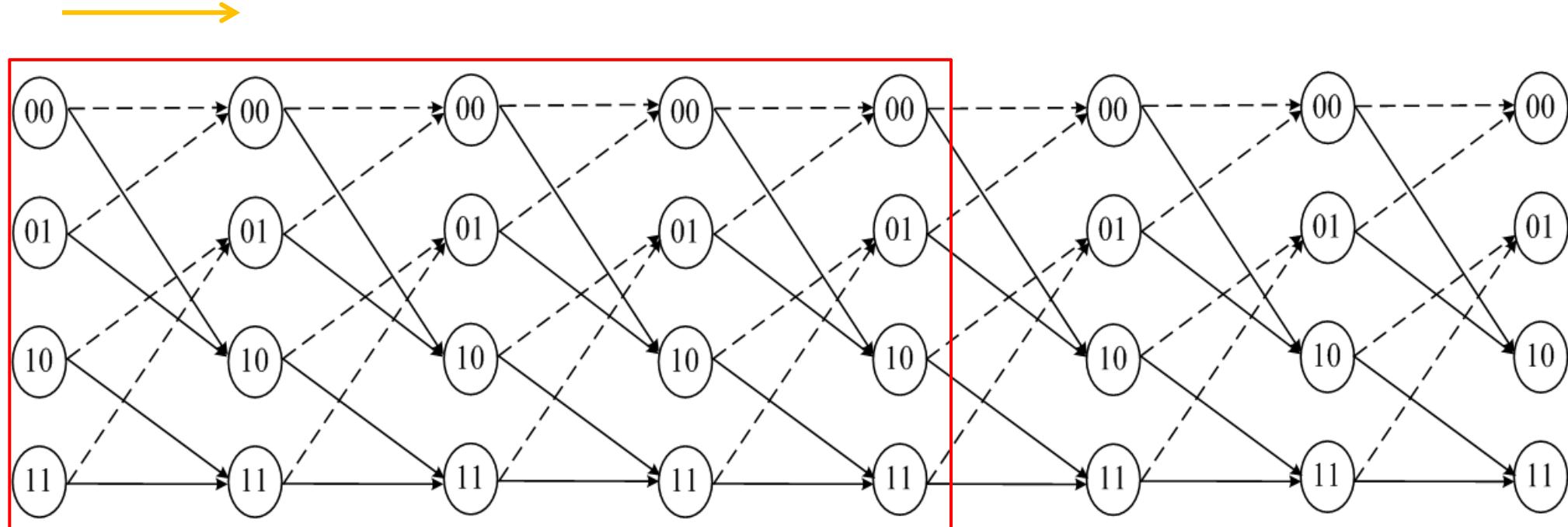
- Junae Li, Shu Lin, Khaled Abdel-Ghaffar, W.E. Ryan, and D.J. Costello, ***LDPC Code Designs, Constructions, and Unification***, Cambridge, 2016
 - C.-C Wong and H.-C Chang, ***Turbo Decoder Architecture for Beyond-4G Applications***, Springer, 2013.
 - W.E. Ryan and Shu Lin, ***Channel codes – Classical and Modem***, Cambridge, 2010
 - R.H. Morelos-Zaragoza, ***The Art of Error Correcting Coding***, 2nd ed., Wiley, 2006
 - Todd K. Moon, ***Error Correction Coding***, Wiley, 2005
 - Shu Lin and D.J. Costello, ***Error-Control Coding***, 2nd ed., Prentice Hall, 2004
 - R.J. McEliece, ***The Theory of Information and Coding***, 2nd ed., Cambridge, 2002.

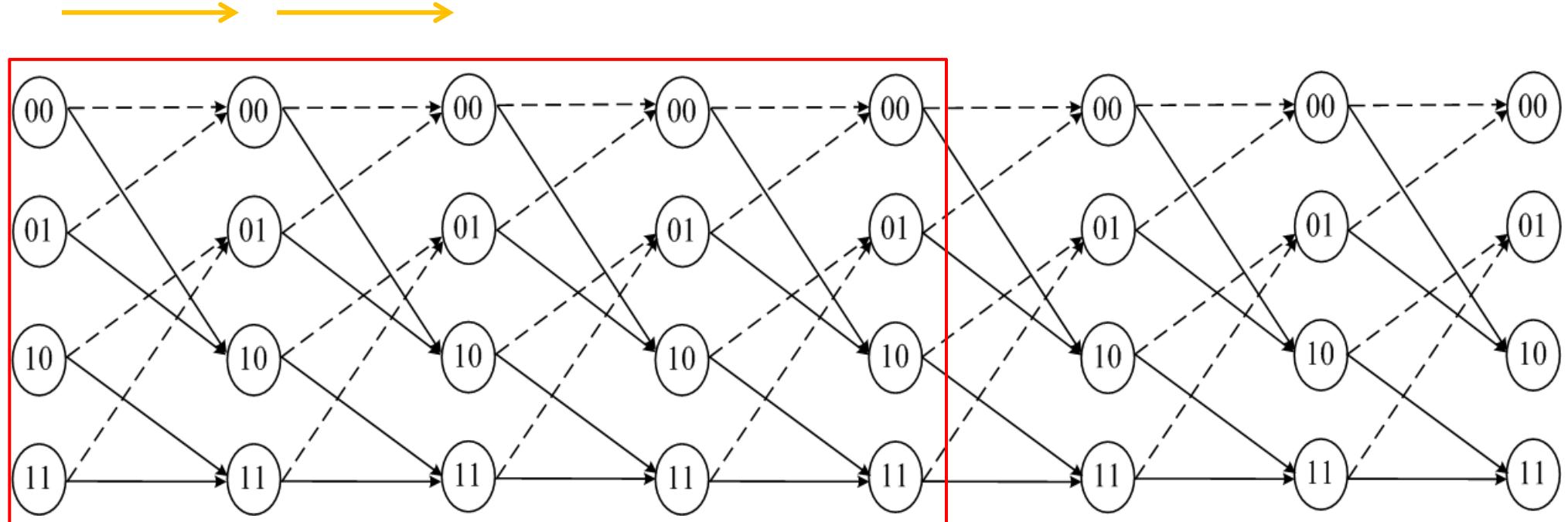
 - “Brief Introduction to Turbo-Like Codes,” ***Ocean group***, NCTU, 2013.
 - “Double Quasi-Cyclic Low-Density Parity Check Codec Design for 5G New Radio,” NTHU, 2017
-

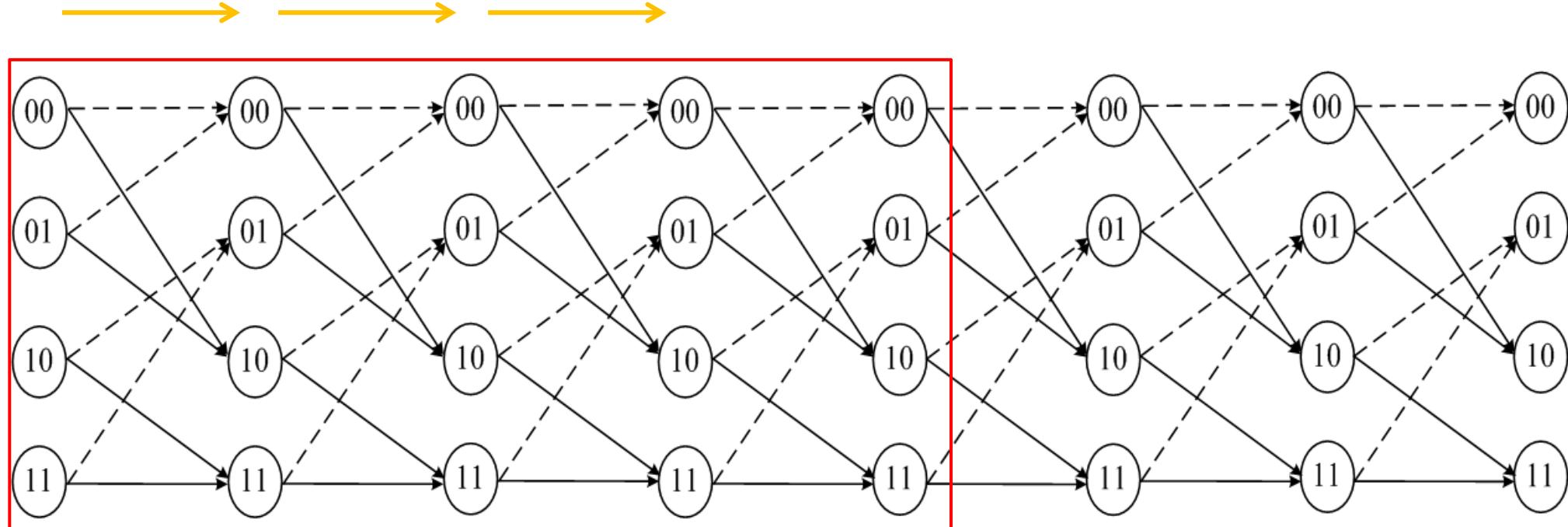


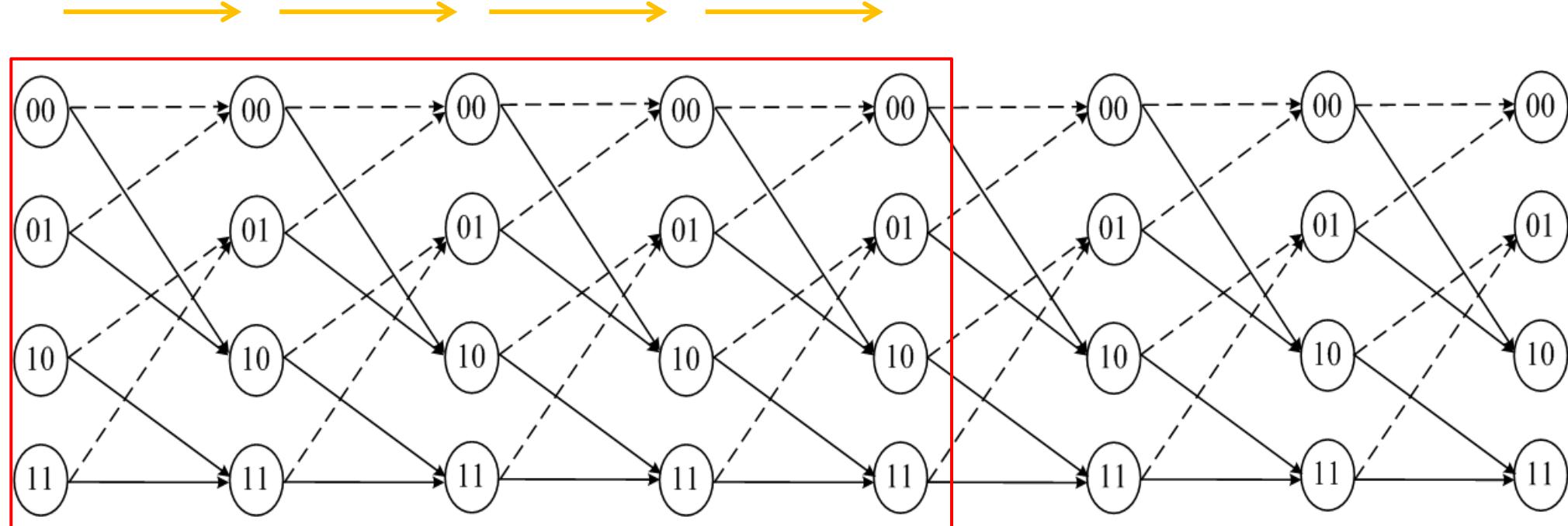
Appendix I - Sliding Window Approach in Trellis

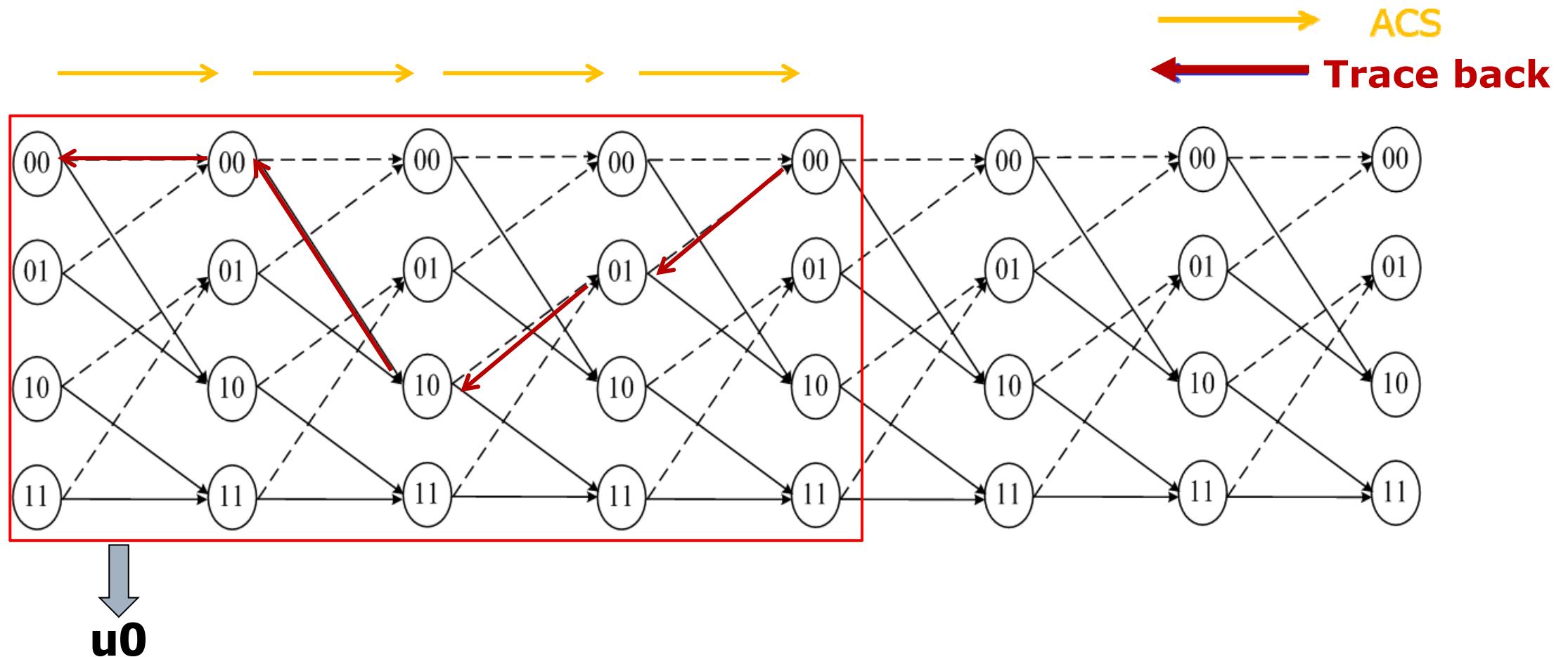


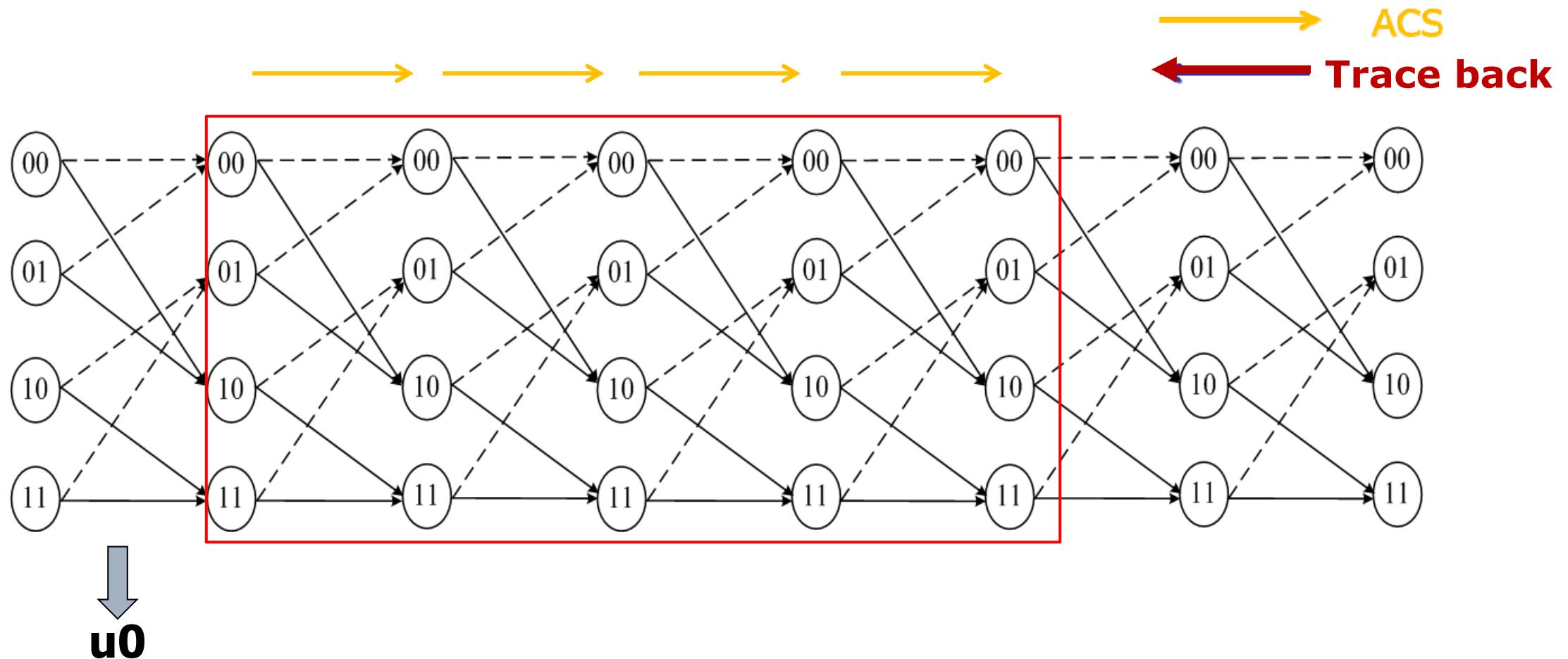


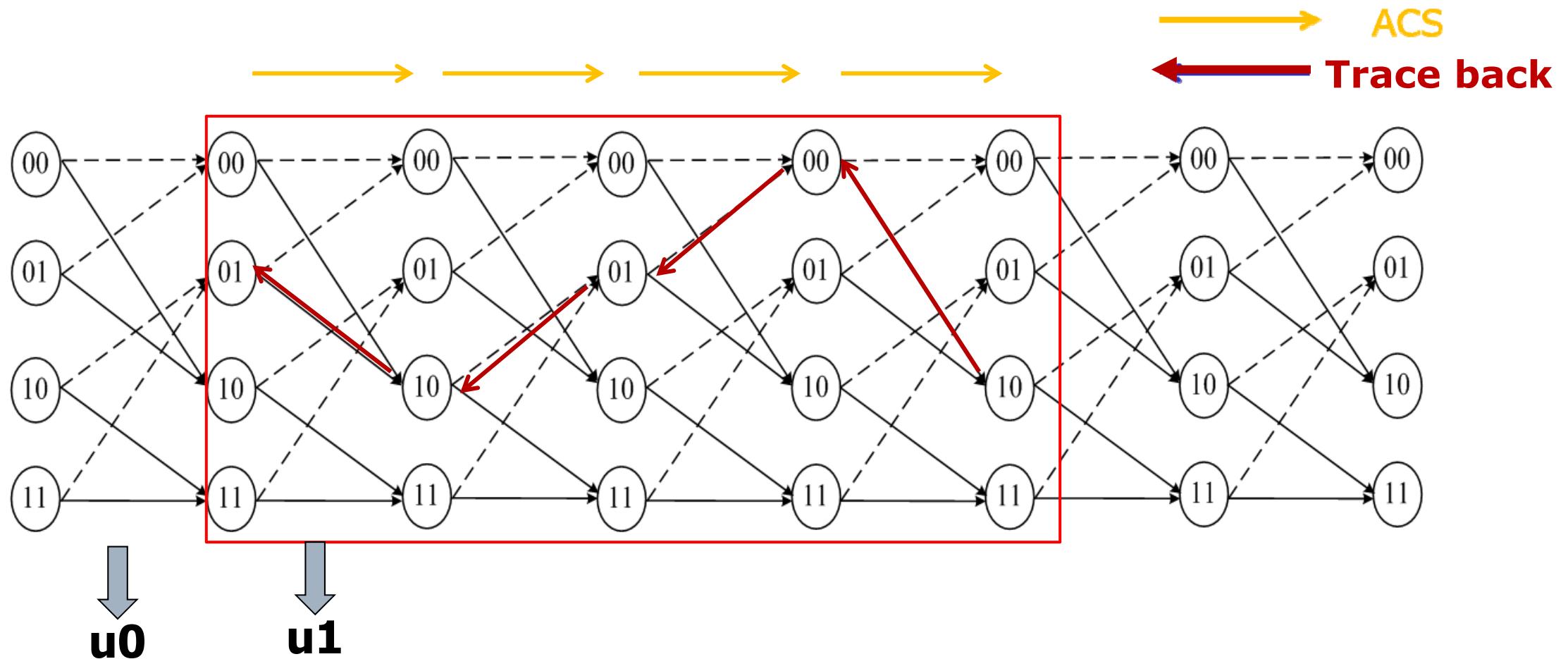




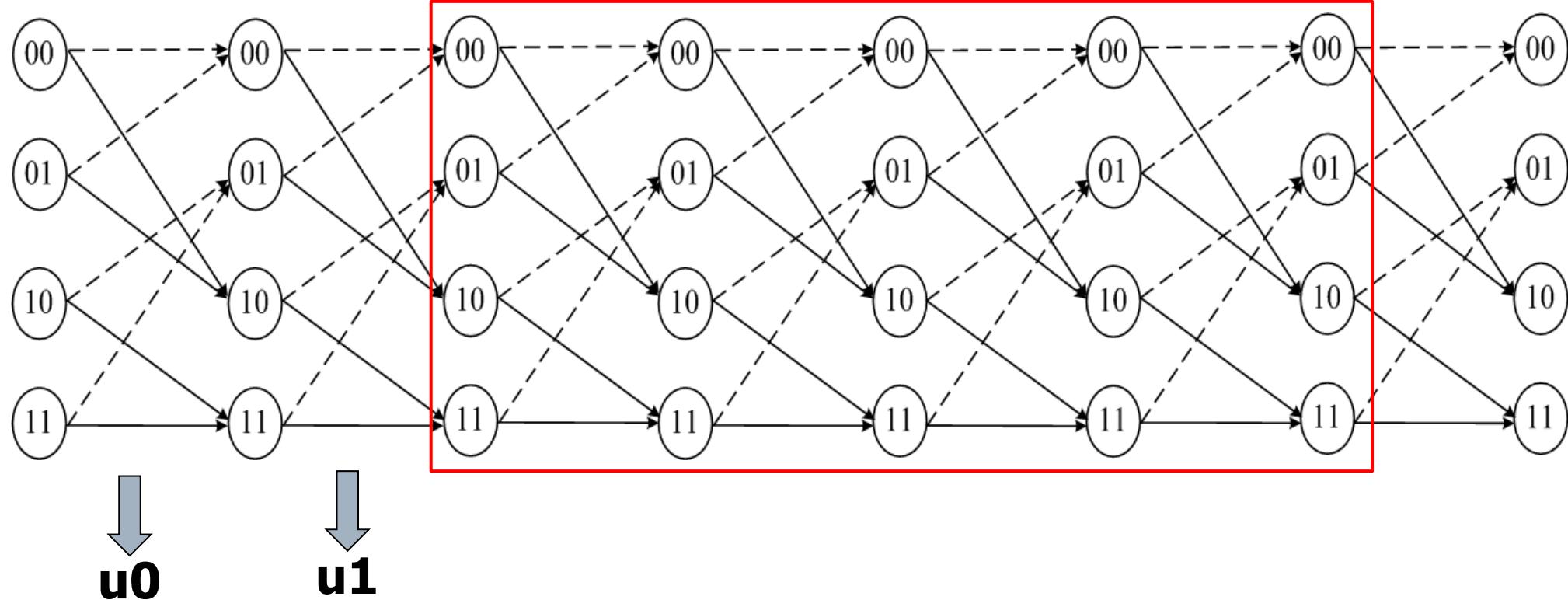




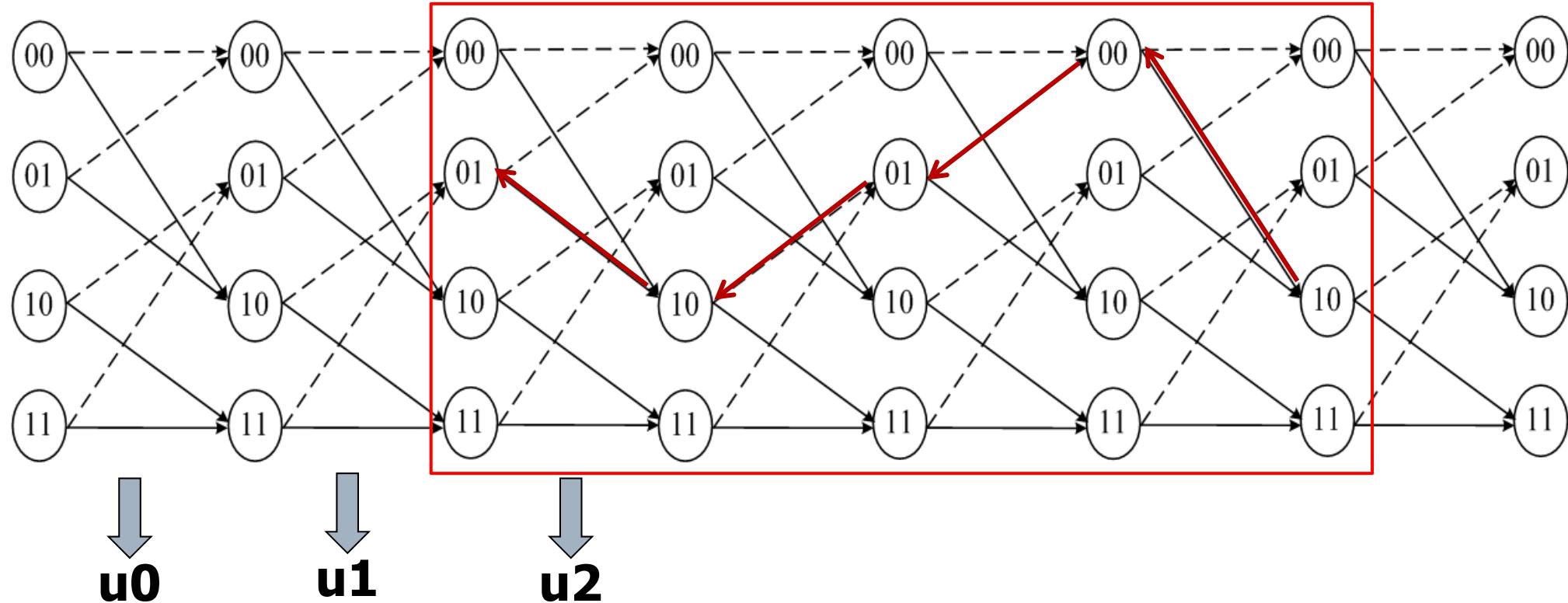




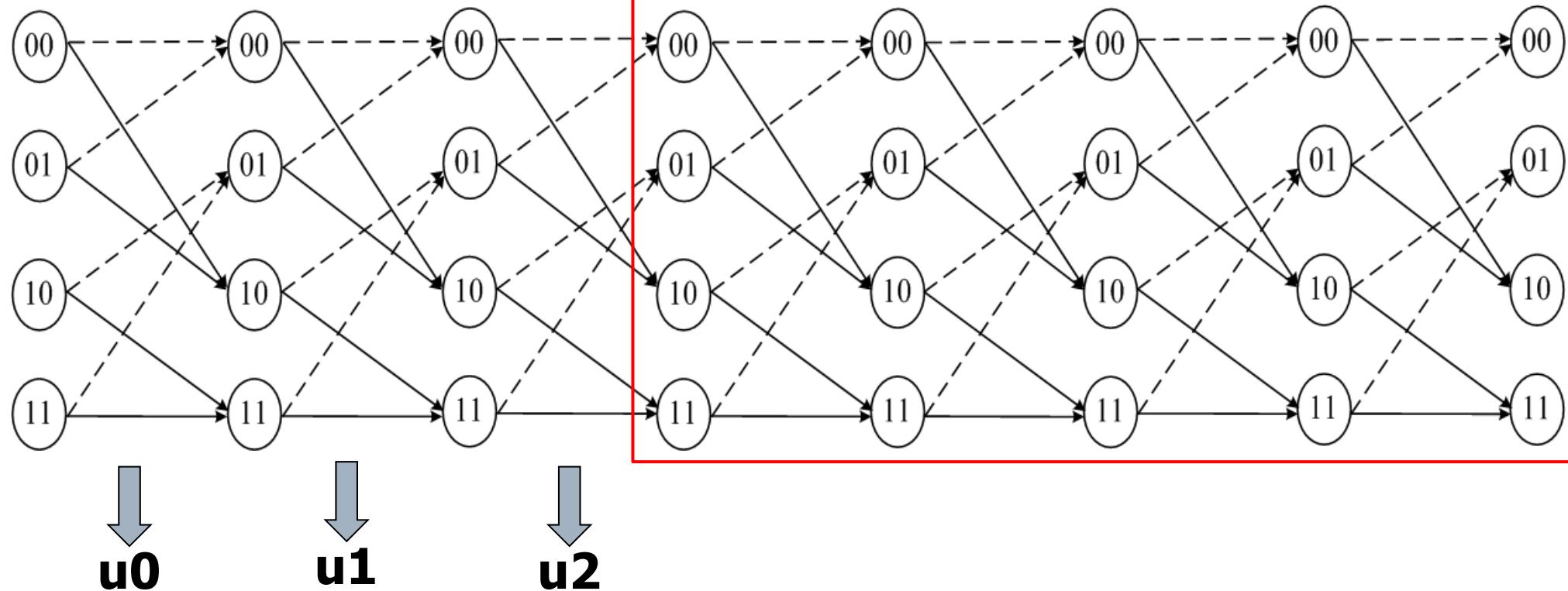
ACS
Trace back



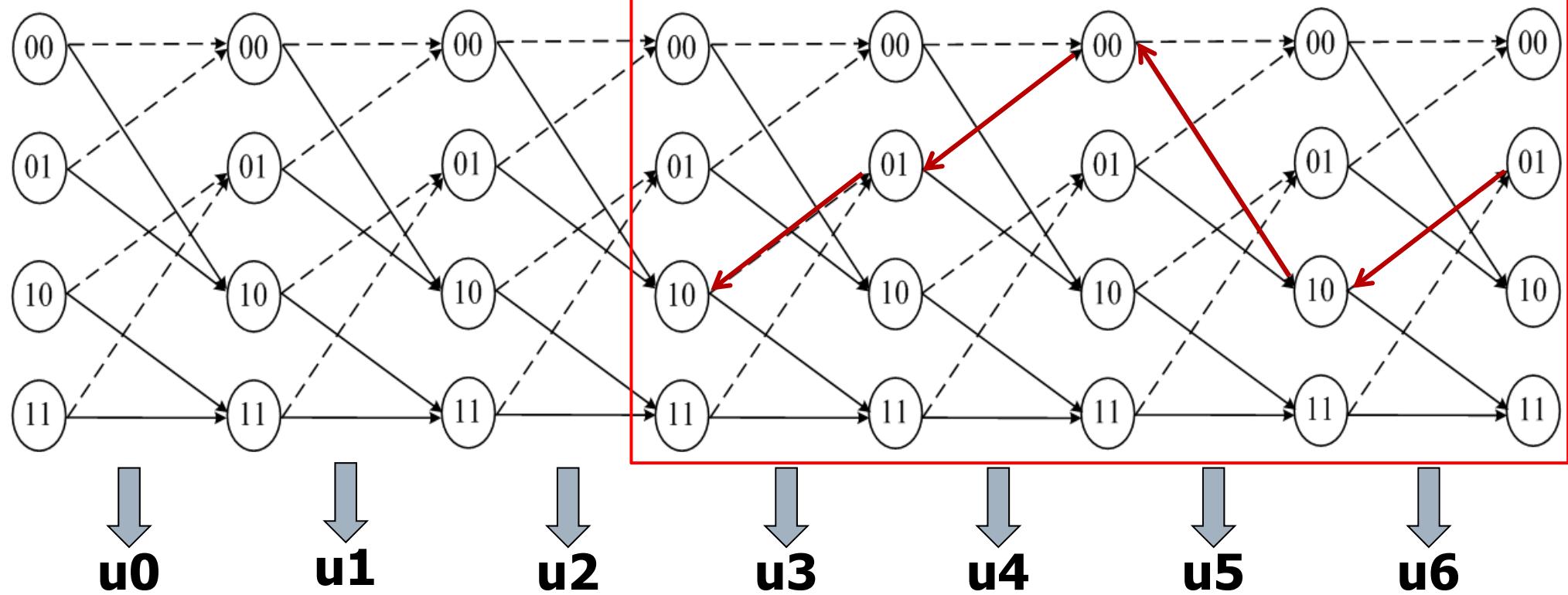
ACS
Trace back



ACS
Trace back



ACS
 Trace back



back

Appendix II – MAP algorithm & Turbo Principle

- Maximum *a posteriori* probability (**MAP**) algorithm
- Decisions u^* with maximum $\Pr(u | r)$

$$\Pr\{r_0, \dots, r_{N-1} \mid y_0, \dots, y_{N-1}\} \triangleq \prod_{i=0}^{N-1} \Pr\{r_i^{(0)}, \dots, r_i^{(n-1)} \mid y_i^{(0)}, \dots, y_i^{(n-1)}\}$$

$$\Pr\{r_i^{(0)}, \dots, r_i^{(n-1)} \mid y_i^{(0)}, \dots, y_i^{(n-1)}\} \triangleq \prod_{j=0}^{n-1} \Pr\{r_i^{(j)} \mid y_i^{(j)}\}$$

$$\Pr\{r_i^{(j)} \mid y_i^{(j)}\} \triangleq \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2} (r_i^{(j)} - y_i^{(j)})^2\right)$$

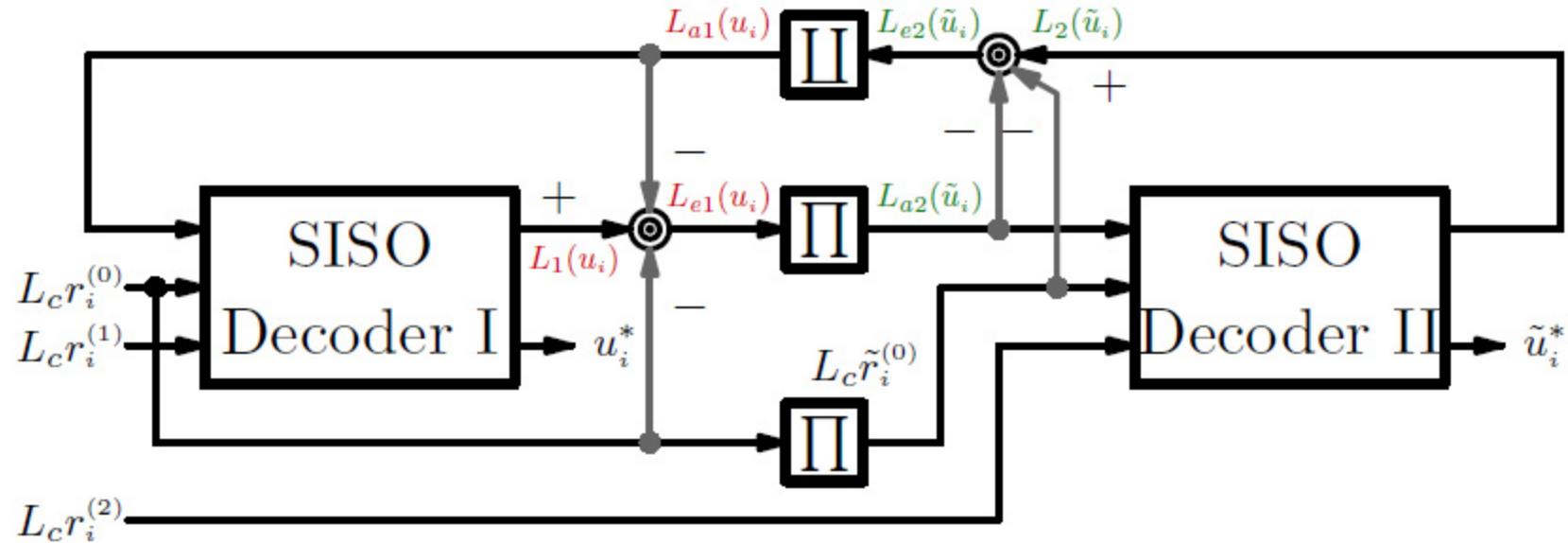
- The *a posteriori* probability of u_i : $\Pr\{u_i \mid r_0, \dots, r_{N-1}\}$

$$u_i^* = \begin{cases} 0 & \text{if } \Pr\{u_i = 0 \mid r_0, \dots, r_{N-1}\} > \Pr\{u_i = 1 \mid r_0, \dots, r_{N-1}\} \\ 1 & \text{otherwise} \end{cases}$$

- The log-likelihood ratio (**LLR**) of u_i :

$$L(u_i) \triangleq \ln \frac{\Pr\{u_i = 0 \mid r_0, \dots, r_{N-1}\}}{\Pr\{u_i = 1 \mid r_0, \dots, r_{N-1}\}} \rightarrow u_i^* = \begin{cases} 0 & \text{if } L(u_i) \geq 0 \\ 1 & \text{if } L(u_i) < 0 \end{cases}$$

□ Turbo Decoder Architecture

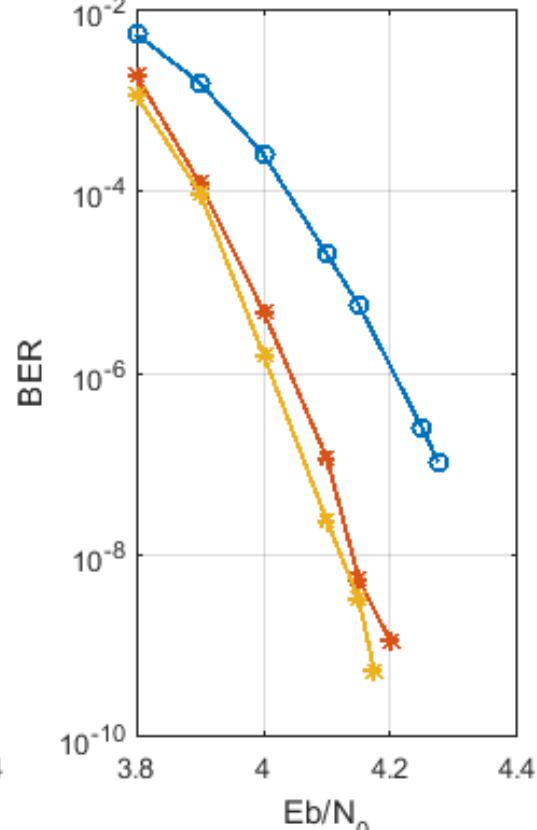
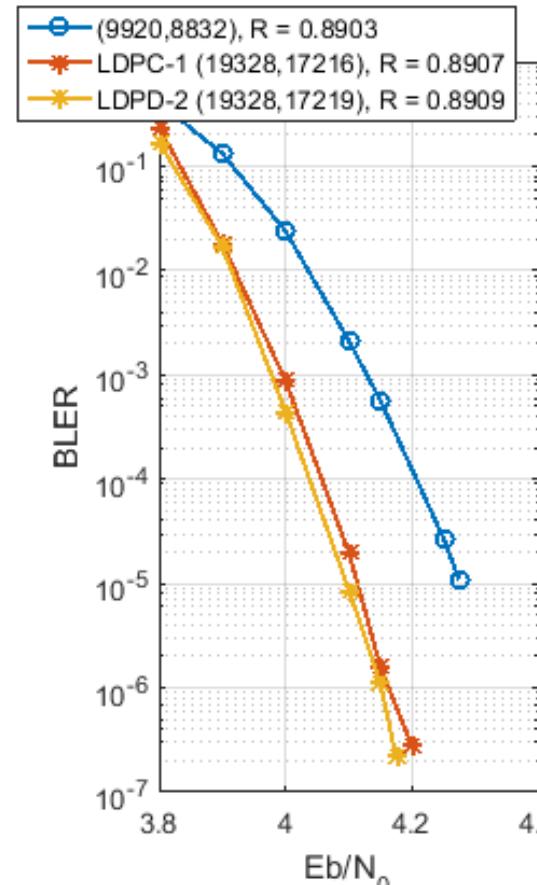
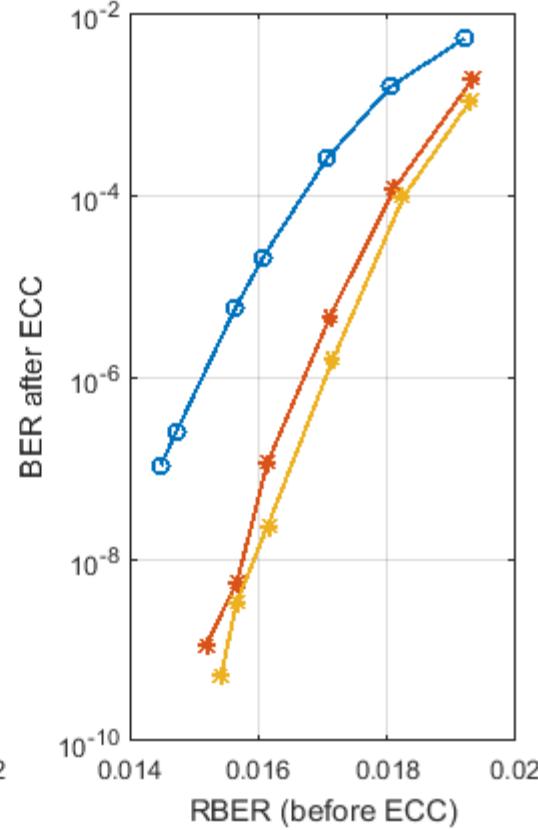
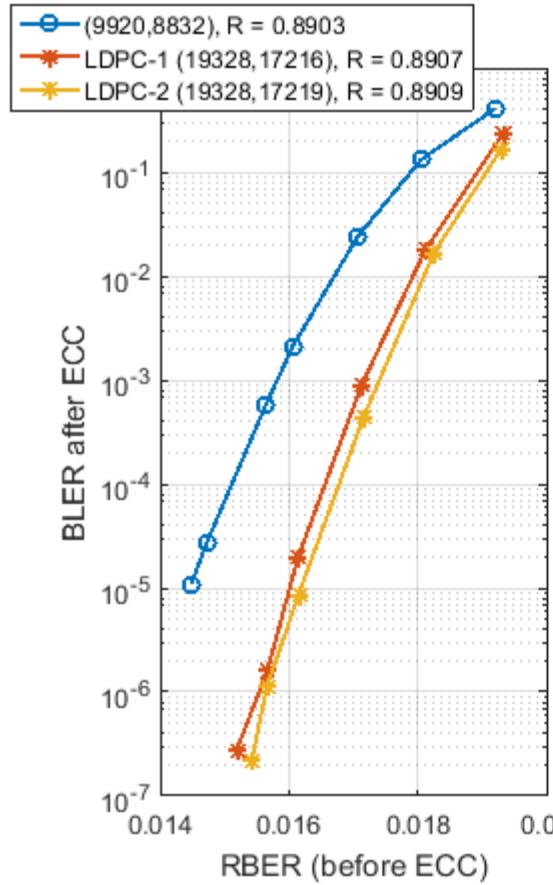


- Extrinsic information calculation

$$\begin{aligned}L_1(u_i) &= L_{a1}(u_i) + L_c r_i^{(0)} + L_{e1}(u_i) \\ \Rightarrow L_{e1}(u_i) &= L_1(u_i) - L_{a1}(u_i) - L_c r_i^{(0)}\end{aligned}$$

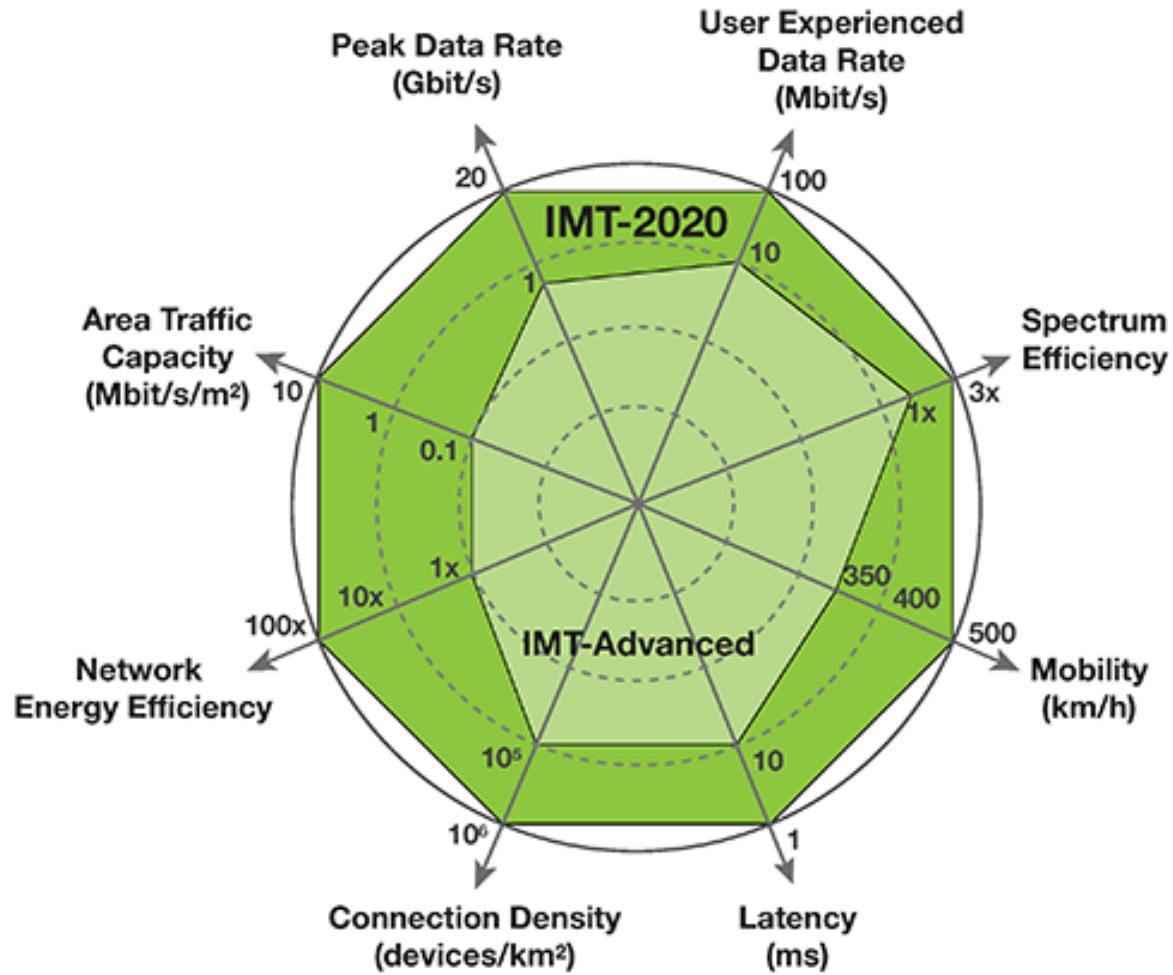
back

Appendix III – 1KB/2KB LDPC Codes Performance



back

What is 5G benefit?



back