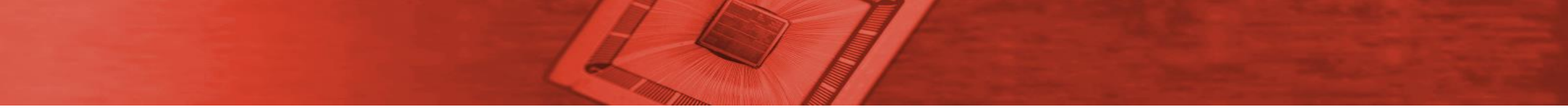


Integrated Circuits for Communications after the Happy Scaling Era

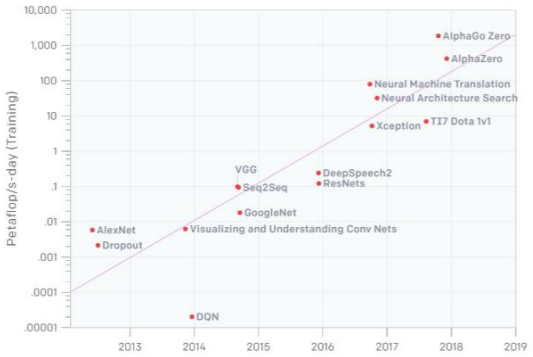
(based on examples from channel coding)

Andreas Burg

Telecommunications Circuits Laboratory, EPFL

The End of the "Happy-Scaling" Era: Lets get to work!

New Applications: AI



Increasing complexity

https://blog.openai.com/ai-and-compute/

More processing in a reduced power envelope

Maintaining progress requires ***design*** innovation

Requirements

Technology Benefits

Arrival of AI

"Happy Scaling" Era

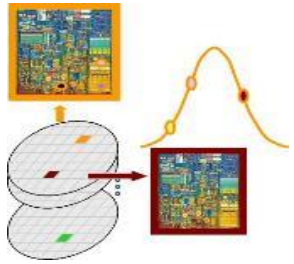
Time

Process

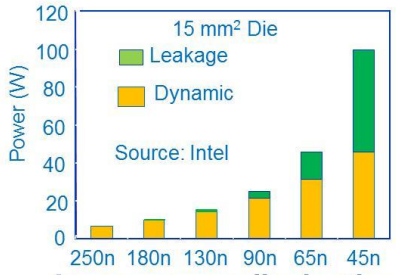
90 nm

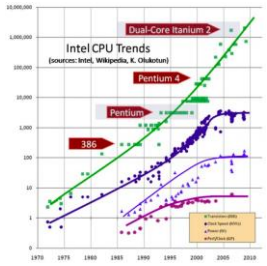
28 nm

TODAY

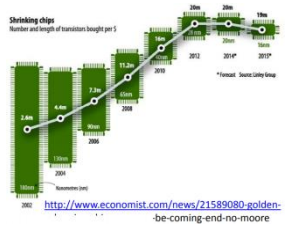


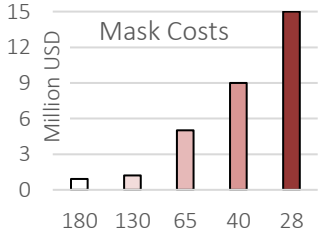
Severe variability and reliability issues





Vanishing energy and performance benefits



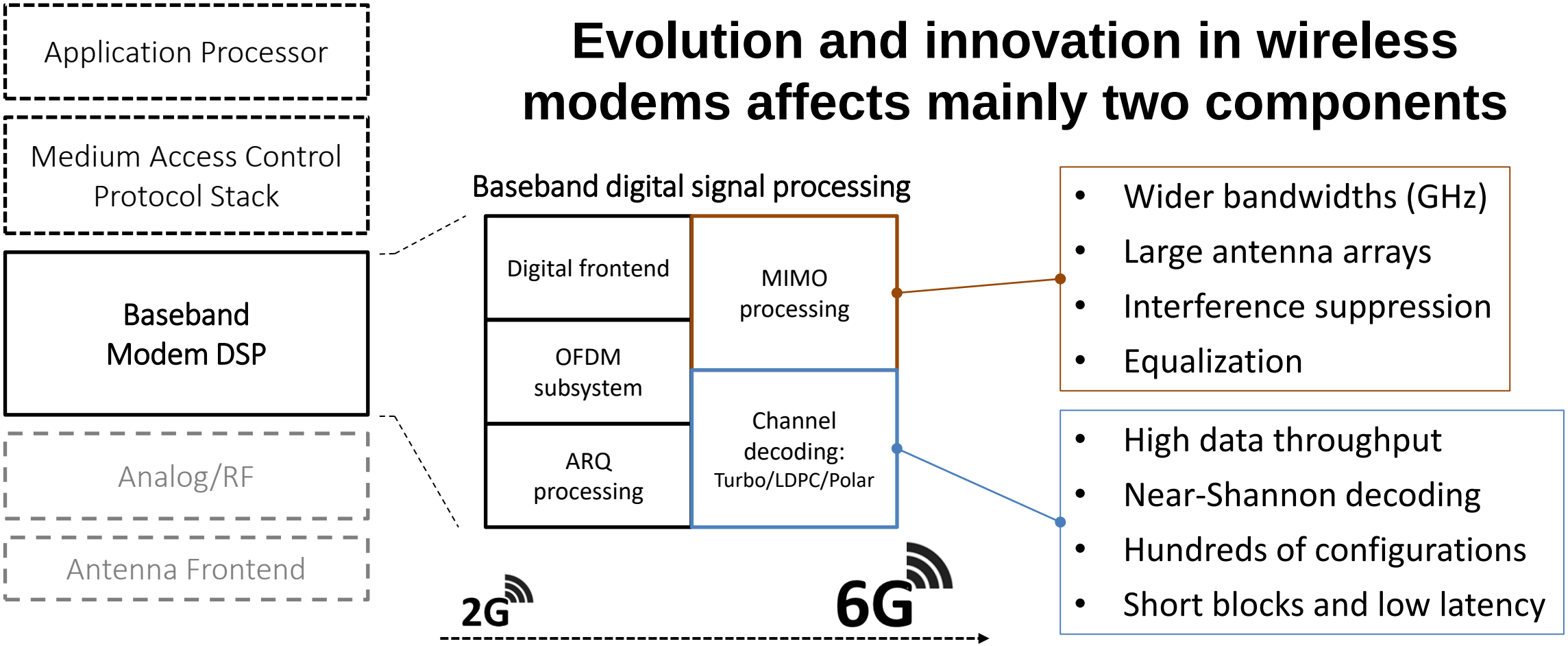


Skyrocketing costs (mask-, silicon, and design-costs)

Wireless Modem Architecture & Trends

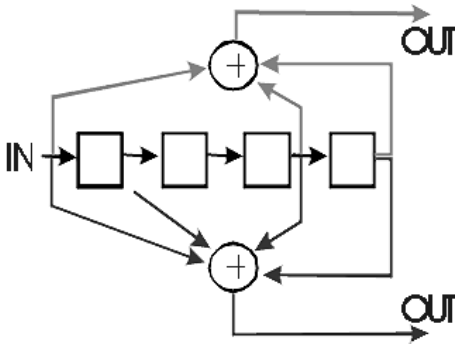
Wireless modem is a key component in mobile SoCs

- Affected by constant updates and innovations as algorithms and standards evolve

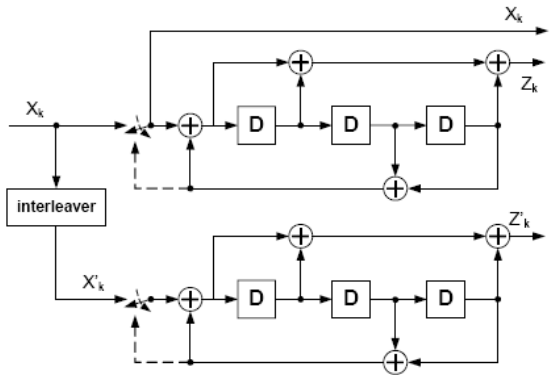


Channel Codes in 3GPP Mobile Wireless Systems

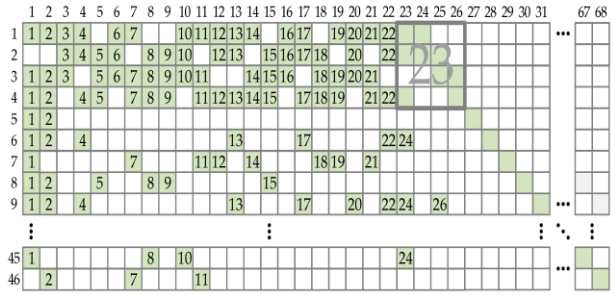
Convolutional Codes



Turbo Codes



LDPC / Polar Codes



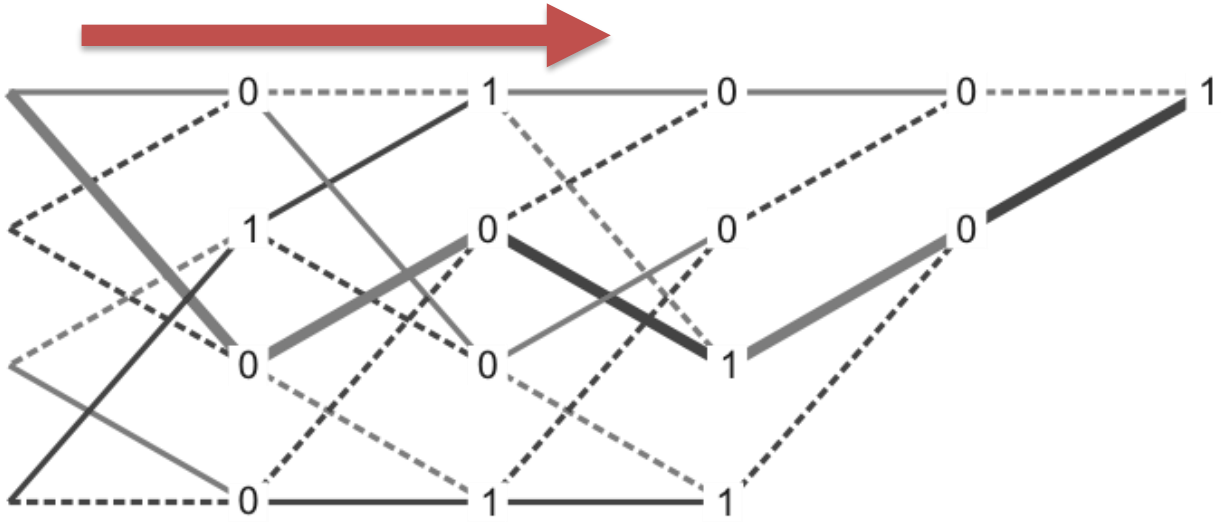
?

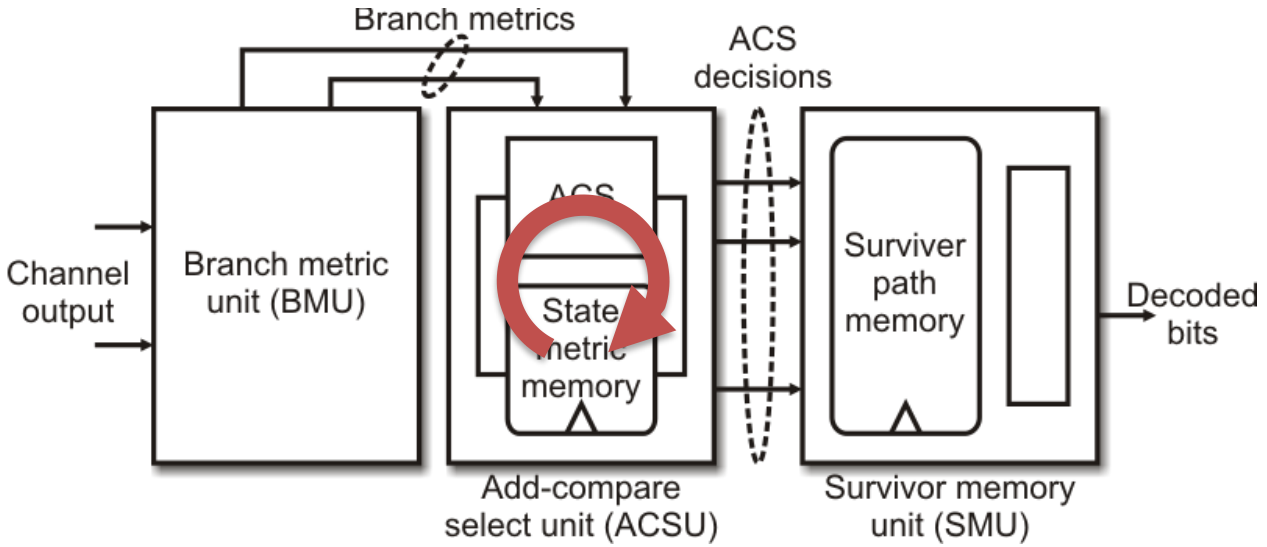
2G 3G 4G 5G 6G

Sequential Decoding of Convolutional Codes

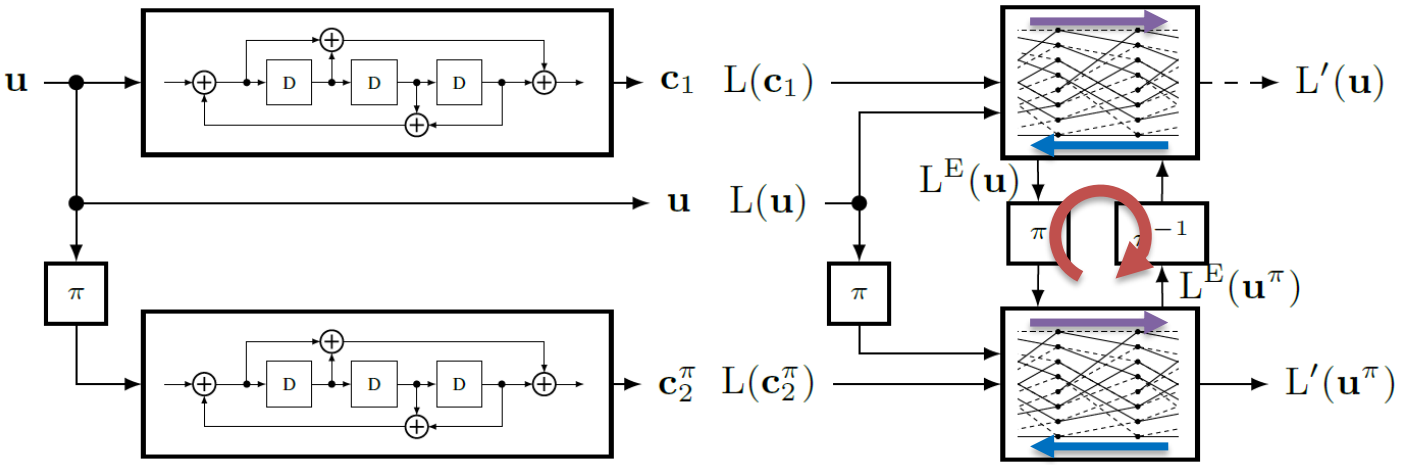
VITERBI algorithm (1967)

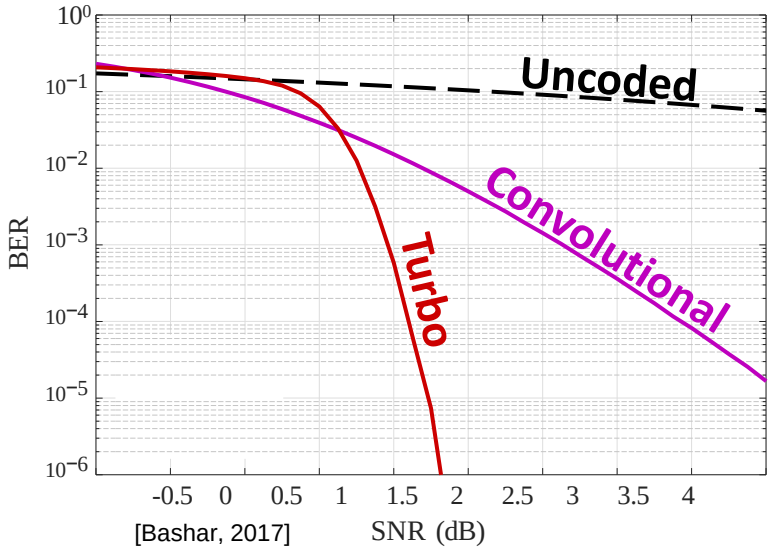
- Step-by-step trellis traversal
- Inherently **sequential process**
  - Bits per clock cycle defined only by the code
  - **Parallel processing** only **with exponential complexity**





3G: From Convolutional to Turbo Codes (1990s)





[Bashar, 2017]

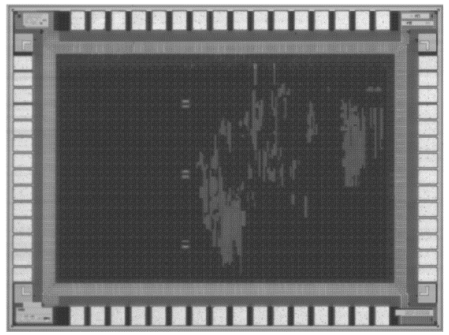
- Encoding with two independent convolutional codes based on "randomly interleaved" data
- Iterative decoding
  - Two component decoders operating on interleaved convolutional codes
  - Exchange of reliability information
  - Component decoders based on the BCJR algorithm
- Performance close to Shannon limit

Tahir, Bashar et al. "BER comparison between Convolutional, Turbo, LDPC, and Polar codes." 2017 24th International Conference on Telecommunications (ICT)

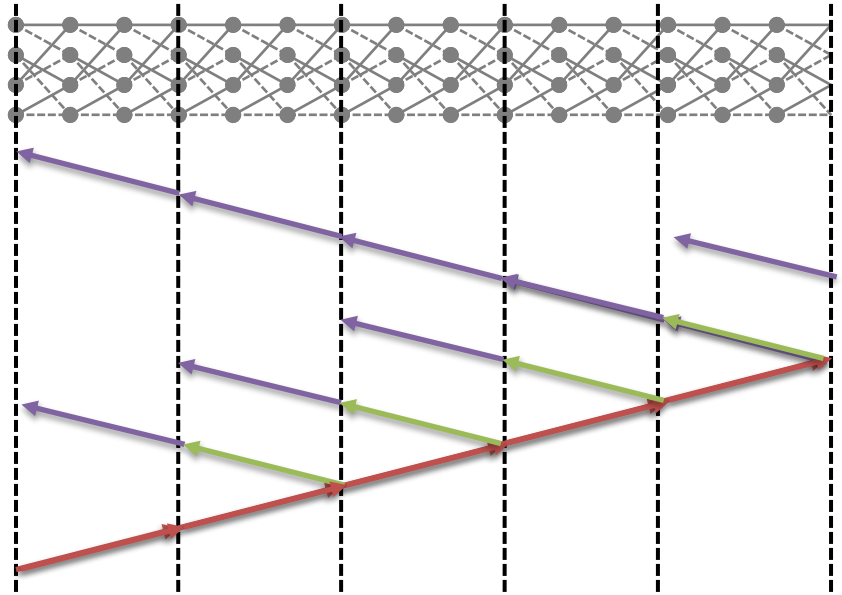
Efficient Decoding of Turbo Codes with Sliding Window BCJR

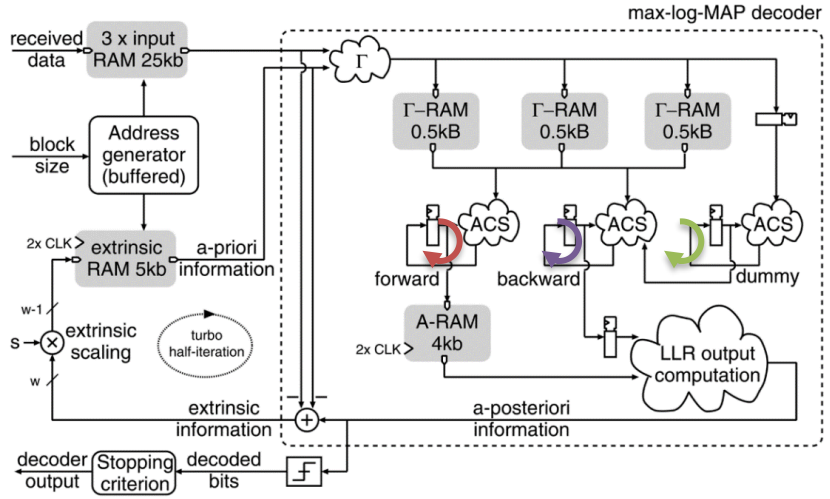
- Structurally **similar to Viterbi algorithm**
- **Requires forward and backward iterations**
  - **Large memory to store intermediate results**
- Decompose trellis into smaller, overlapping windows
  - Memory reduction at the cost of additional complexity
  - Forward, backward, and dummy iterations can be parallelized: less memory and higher throughput

130nm CMOS

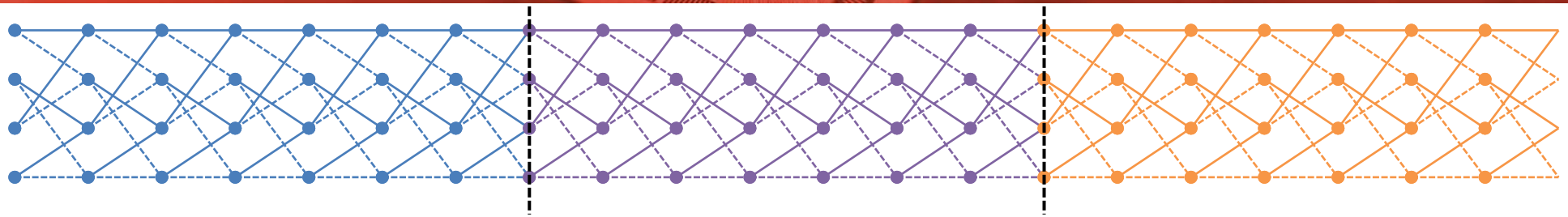


Throughput	**390 Mbps**
Energy/bit	**~2nJ/bit**



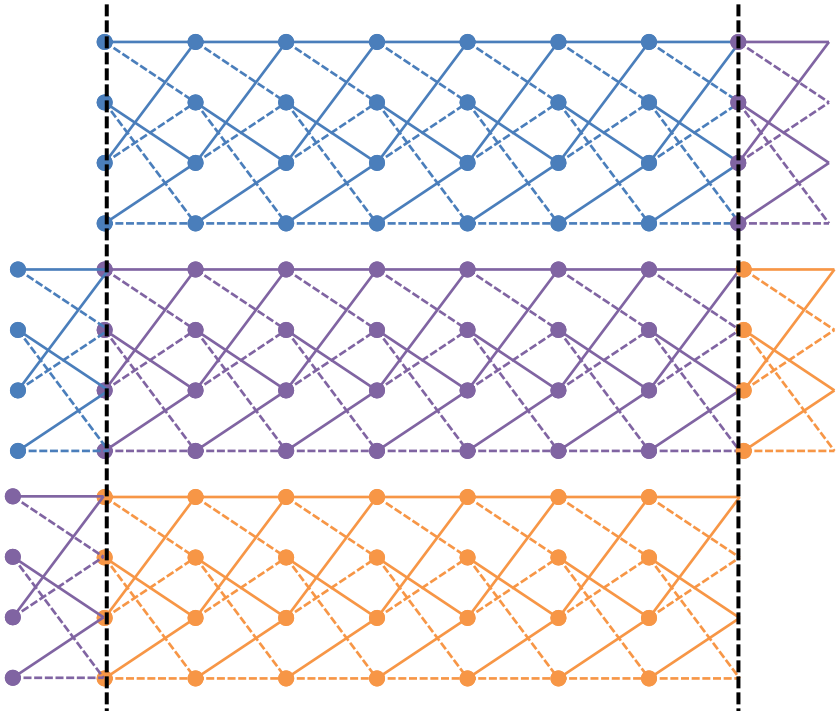


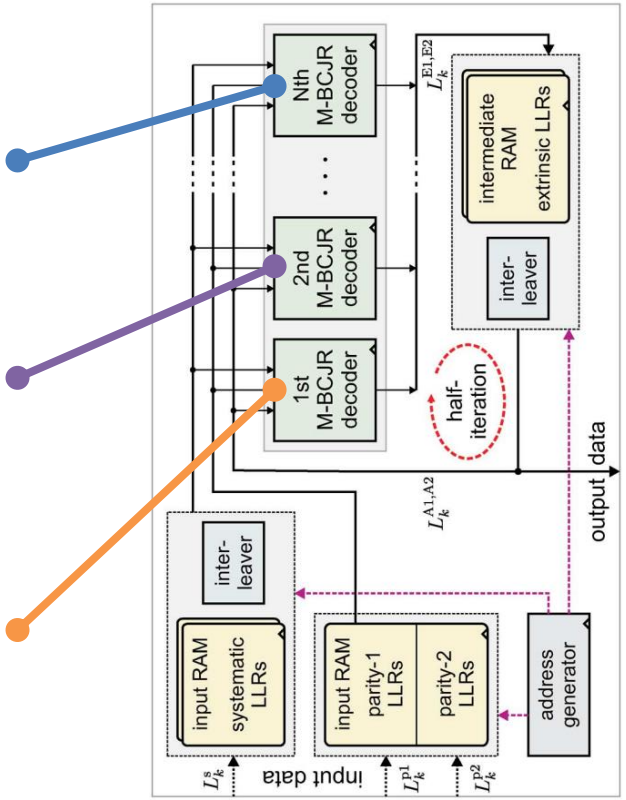
Boosting the throughput of BCJR: Parallel Sliding Window



Process multiple trellis **windows in parallel** with parallel BCJRs

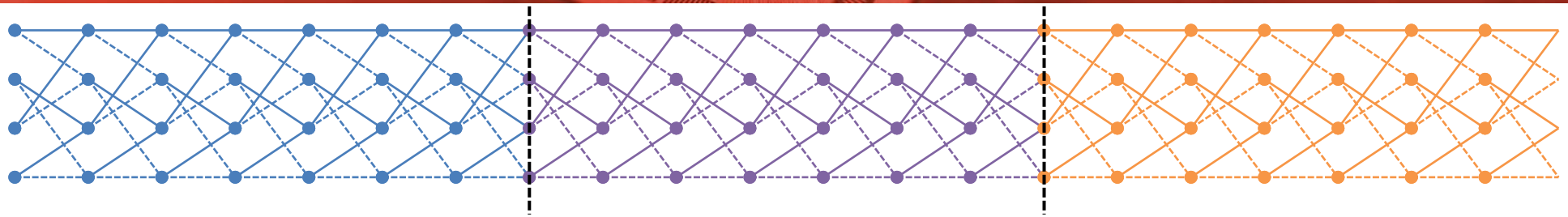
- Parallel memory access
  - Even iterations (non-interleaved): no access conflicts





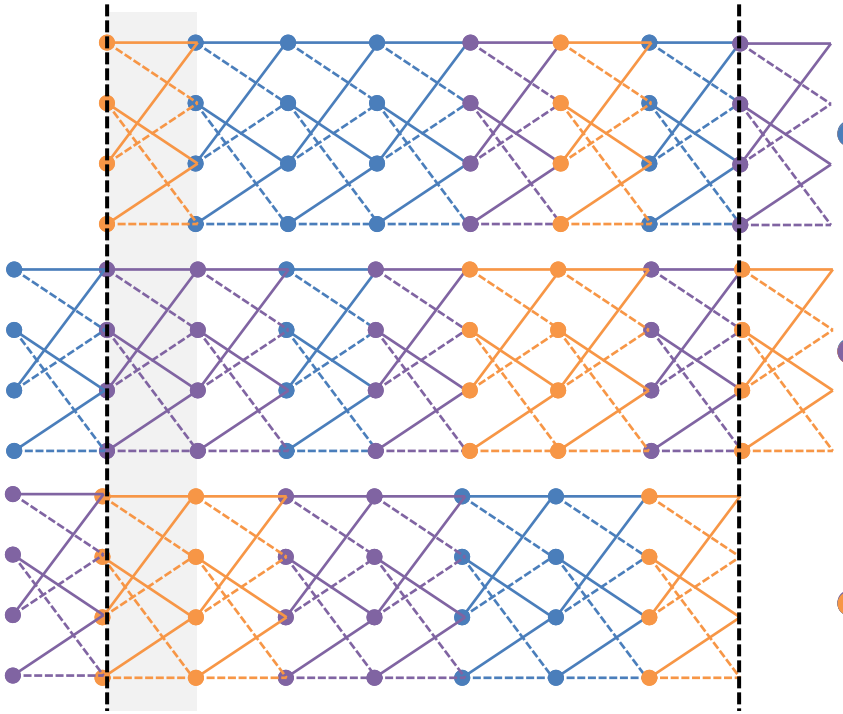
[Studer, 2011]

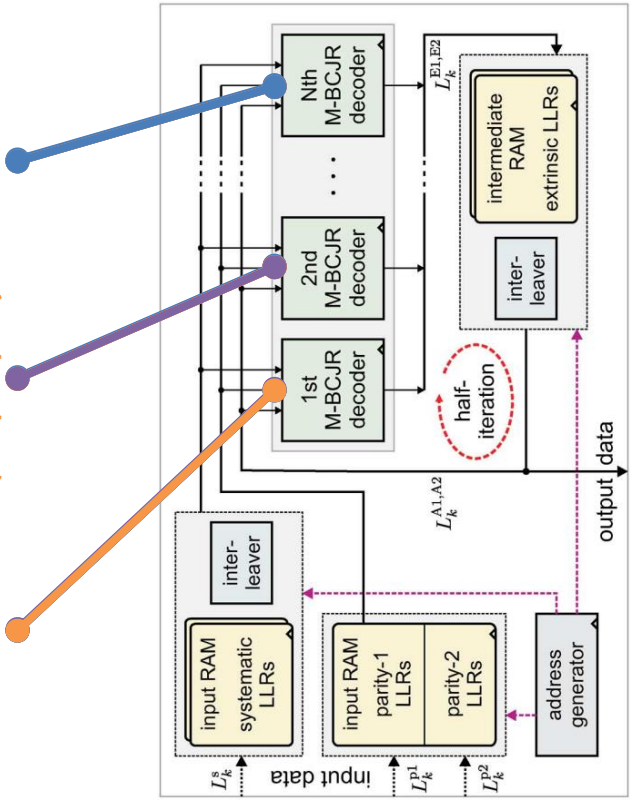
Boosting the throughput of BCJR: Parallel Sliding Window



Process multiple trellis **windows in parallel** with parallel BCJRs

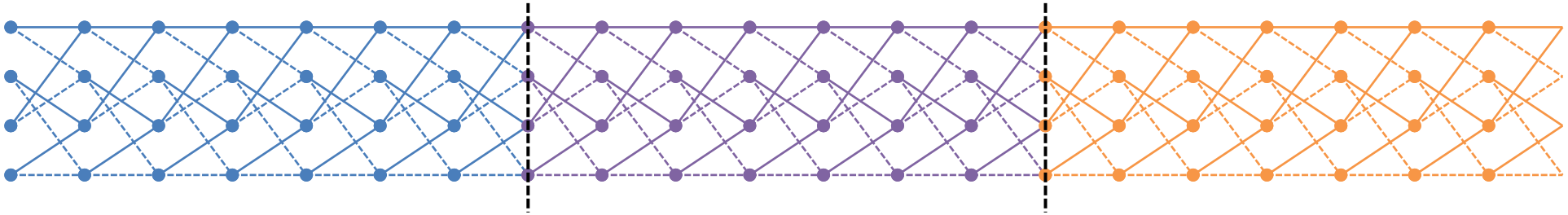
- Parallel memory access
  - Non-interleaved iterations: no access conflicts
  - **Interleaved iterations: access conflicts** due to re-ordering





[Studer, 2011]

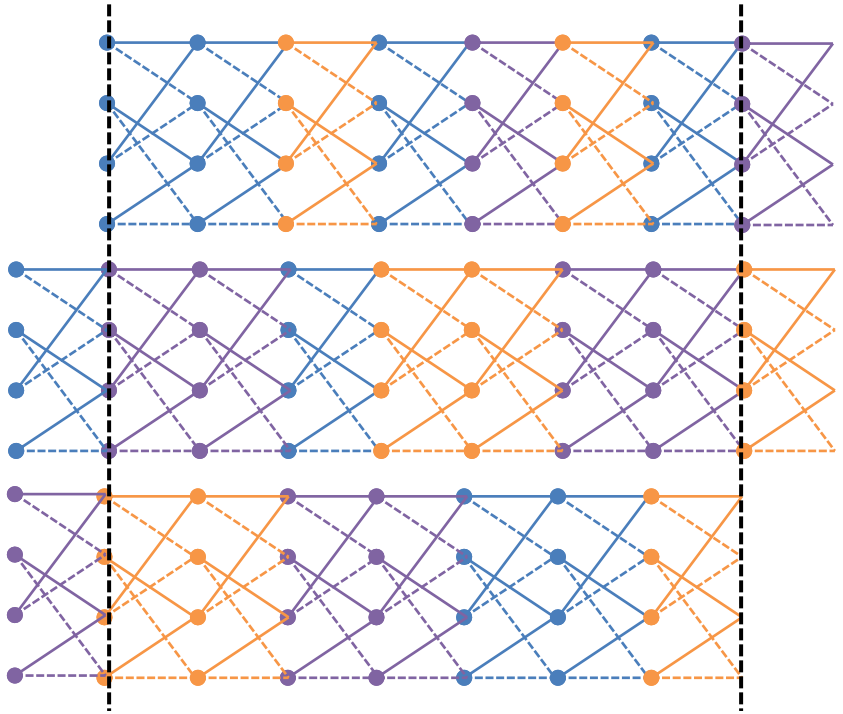
Boosting the throughput of BCJR: Parallel Sliding Window

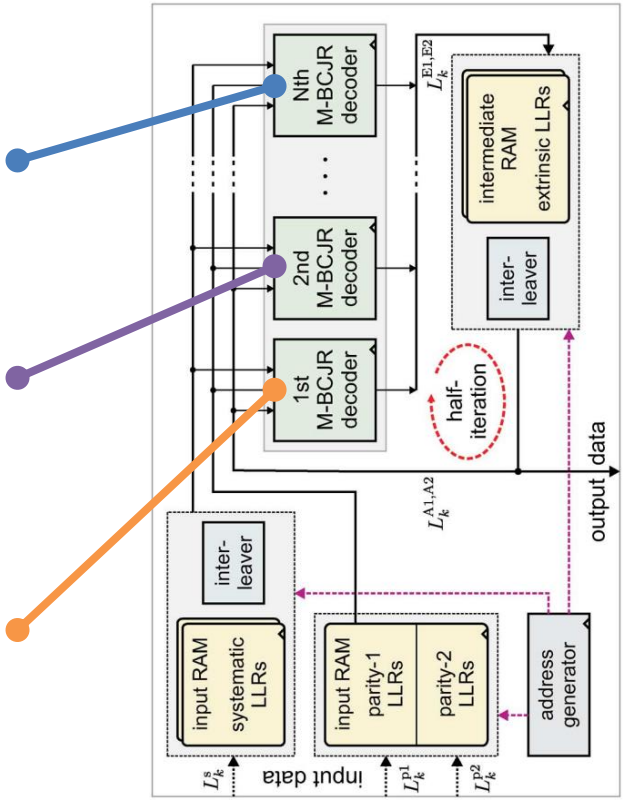


Process multiple trellis **windows in parallel** with parallel BCJRs

- Parallel memory access
  - Non-interleaved iterations: no access conflicts
  - **Interleaved iterations: access conflicts** due to re-ordering

- **4G-LTE** defines a hardware-optimized Quadratic-permutation-polynomial (QPP) interleaver that avoids access conflicts

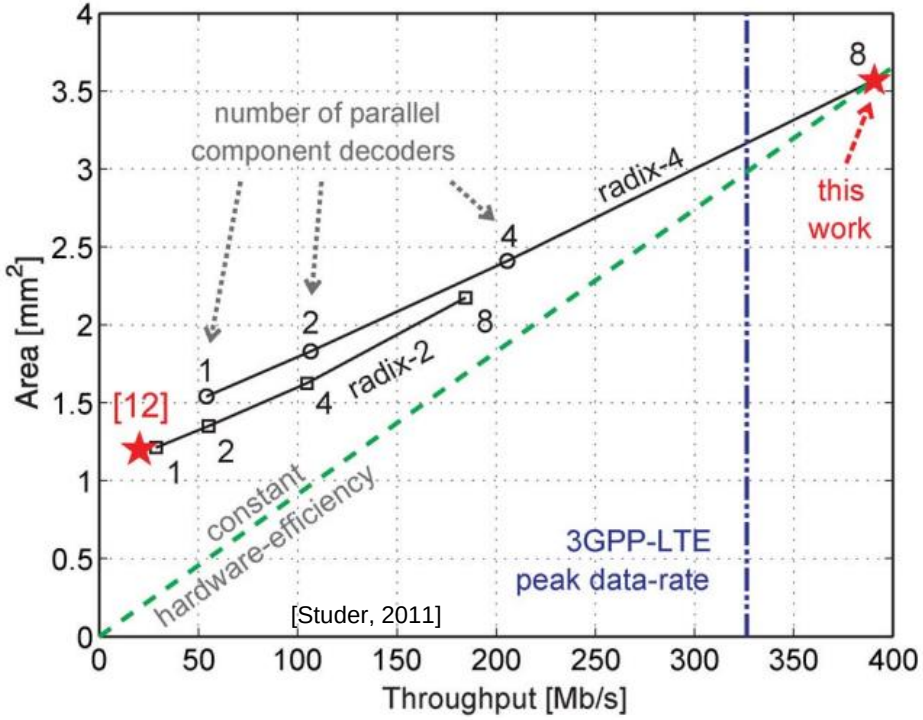


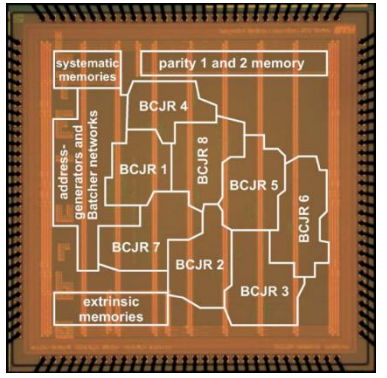


[Studer, 2011]

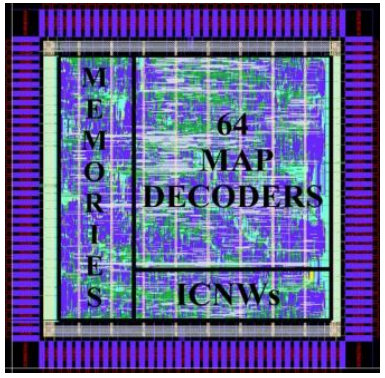
Boosting the throughput of BCJR: Parallel Sliding Window

- **Memory consumes** a **significant** portion of the **area**
  - More parallel functional units lead to better hardware-efficiency
- Maximum parallelism limited by the interleaver
  - increasing structure eventually impacts FER performance
- Reference implementations reach to **only ~1 Gbps even in advanced technology nodes**





[Studer, 2011]



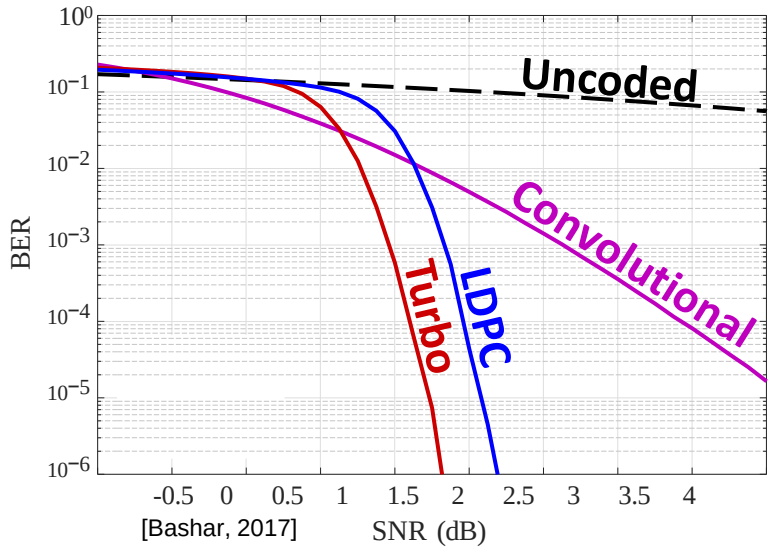
[R. Shrestha, 2014]

C. Studer, C. Benkeser, S. Belfanti and Q. Huang, "Design and Implementation of a Parallel Turbo-Decoder ASIC for 3GPP-LTE," in IEEE Journal of Solid-State Circuits, vol. 46, no. 1, pp. 8-17, Jan. 2011

R. Shrestha and R. P. Paily, "High-Throughput Turbo Decoder With Parallel Architecture for LTE Wireless Communication Standards," in IEEE Transactions on Circuits and Systems I: Regular Papers, 2014

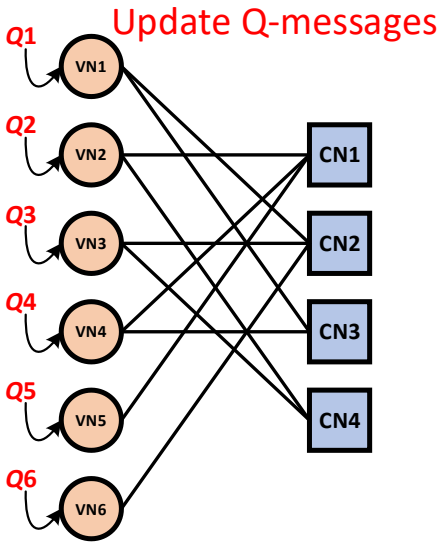
The Comeback of a Forgotten Idea: Low Density Parity Check (LDPC)

- **Discovered by Gallager already in 1963**
- Linear block code with a sparse parity check matrix
- Performance close to Shannon limit
- Belief Propagation (BP) Decoding
  - **Passing messages on a** factor **graph**
  - **Highly parallel**
  - Many **short component codes** (CNs)
- **Deemed to complex for implementation**
- **Come-back in the wake of VLSI parallelism (MacKey, 1996)**



$$H \begin{bmatrix} i \\ p \end{bmatrix} = 0$$

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

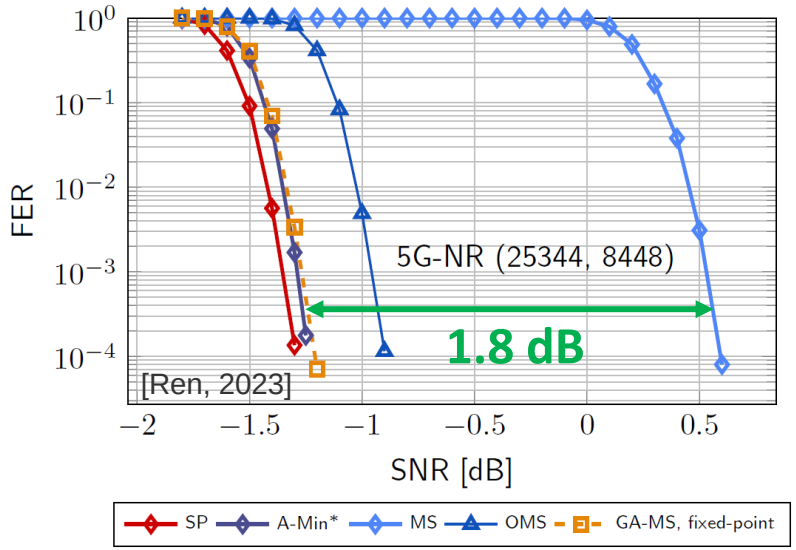


MIN-SUM and Friends: Many Approximations are Good Enough

$$R_{c,v} = \tanh^{-1}\left\{\prod_{n\prime\in\mathcal{N}(m)\backslash \mathrm{n}} \tanh\left(T_{v,c}/2\right)\right\}$$

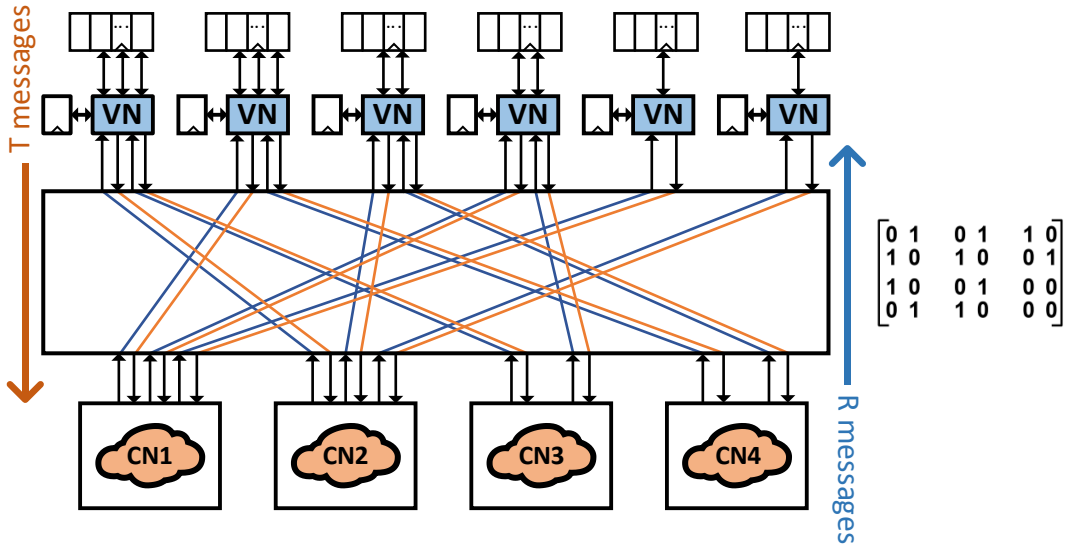
Optimal Sum-Product BP decoding has prohibitively complex message update rules

Sum Product (SP)	$R_{c,v} = \prod_{v'\in M(c)\backslash v} \mathrm{sgn}(T_{v',c}) \cdot \boxplus_{v'\in M(c)\backslash v} \lvert T_{v',c}\rvert$	F. R. Kschischang, *et al, TIT, 2001*
Min-Sum (MS)	$R_{c,v} = \prod_{v'\in M(c)\backslash v} \mathrm{sgn}(T_{v',c}) \cdot \min_{v'\in M(c)\backslash v} \lvert Q_{v',c}\rvert$	N. Wiberg, *1996*
Approximate Min (A-Min*)	$R_{c,v} = \begin{cases} \prod_{v'\in M(c)\backslash v} \mathrm{sgn}(T_{v',c}) \cdot \boxplus_{v'\in M(c)\backslash v} \lvert T_{v',c}\rvert, \text{if } T_{v,c} \text{ is the minimum,} \\ \prod_{v'\in M(c)\backslash v} \mathrm{sgn}(T_{v',c}) \cdot \boxplus_{v\in M(c)} \lvert T_{v,c}\rvert, \text{otherwise.} \end{cases}$	C. Jones, *et al, 2003*
Normalized MS (NMS)	$R_{c,v} = \alpha \cdot \prod_{v'\in M(c)\backslash v} \mathrm{sgn}(T_{v',c}) \cdot \min_{v'\in M(c)\backslash v} \lvert T_{v',c}\rvert$	J. Chen, *et al, TCOM, 2005*
Offset MS (OMS)	$R_{c,v} = \prod_{v'\in M(c)\backslash v} \mathrm{sgn}(T_{v',c}) \cdot \max\left(\min_{v'\in M(c)\backslash v} \lvert T_{v',c}\rvert - \beta, 0\right)$	J. Chen, *et al, TCOM, 2005*
Self Correction MS	$T_{v,c} = 0, \text{if } \mathrm{sgn}\left(T_{v,c}^{(i)}\right) \neq \mathrm{sgn}\left(T_{v,c}^{(i-1)}\right)$	V. Savin, *et al, ISIT, 2008*
Adjusted MS (A-MS)	$R_{c,v} = \begin{cases} \prod_{v'\in M(c)\backslash v} \mathrm{sgn}(T_{v',c}) \cdot \mathrm{LUT}_{v'\in M(c)\backslash v} \lvert T_{v',c}\rvert, \text{if } T_{v,c} \text{ is the minimum,} \\ \prod_{v'\in M(c)\backslash v} \mathrm{sgn}(T_{v',c}) \cdot \mathrm{LUT}_{v\in M(c)} \lvert T_{v,c}\rvert, \text{otherwise.} \end{cases}$	T. Richardson, *et al, 2018*
Adaptive MS	$R_{c,v} = \begin{cases} \prod_{v'\in M(c)\backslash v} \mathrm{sgn}(T_{v',c}) \cdot \max\left(\min_{v'\in M(c)\backslash v} \lvert T_{v',c}\rvert - \beta, 0\right), \text{if } c \leq 4 \\ \prod_{v'\in M(c)\backslash v} \mathrm{sgn}(T_{v',c}) \cdot \min_{v'\in M(c)\backslash v} \lvert T_{v',c}\rvert, \text{otherwise} \end{cases}$	K. Trung, *et al, ISCAS, 2019*



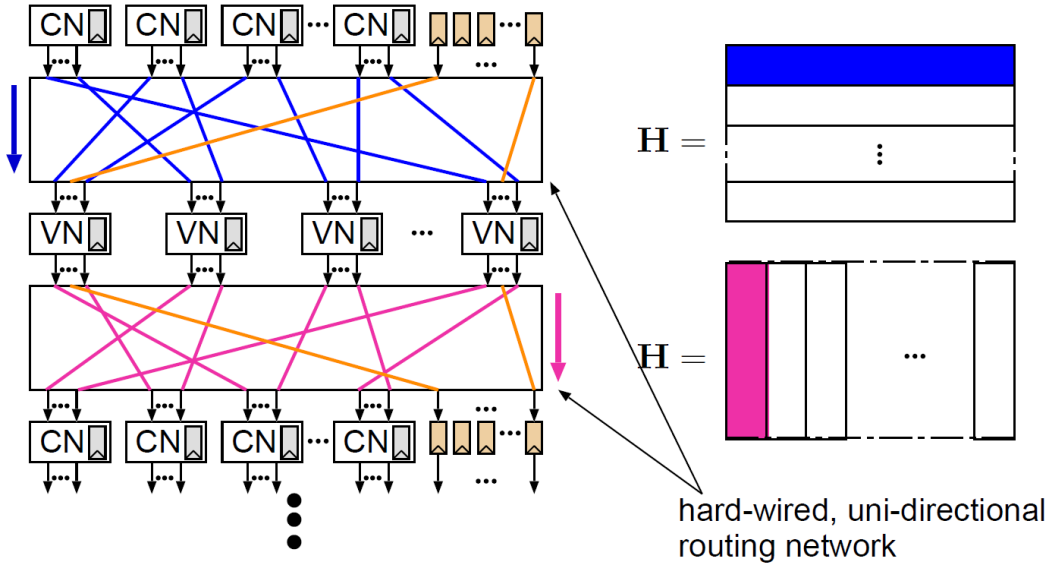
5G-NR LDPC Decoder	MS	GA-MS
T/P [Gbps]	29.40	28.25 **(-3.9%)**
Area [mm2]	1.24	1.33 **(+7.2%)**

Scaling (More Transistors) Enable Extremely High Throughput



Isomorphic architecture

Direct mapping of Tanner graph onto silicon



Pipelined isomorphic architectures

Parallelize iterations: one cycle per code word

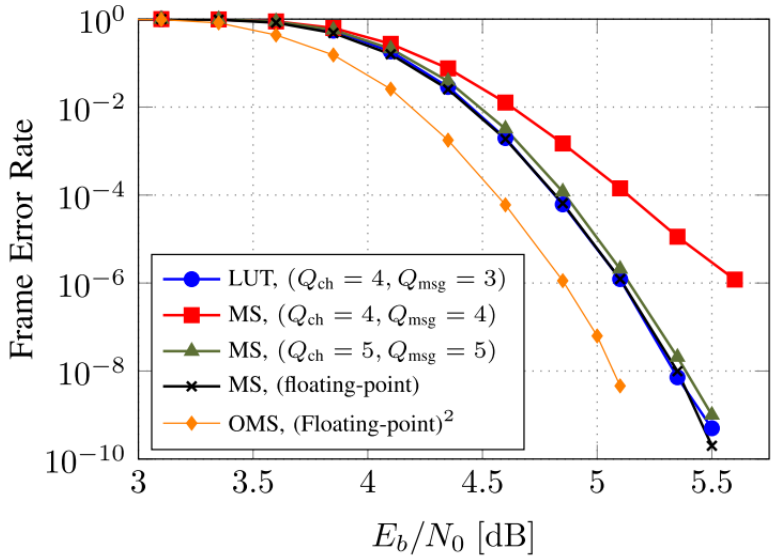
Bottleneck: Message exchange between VNs and CNs

Routing of messages between layers leads to extreme routing congestion

Only feasible for codes with low row/column degrees (poor performance)

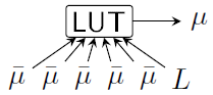
Finite Alphabet Decoding to the Rescue

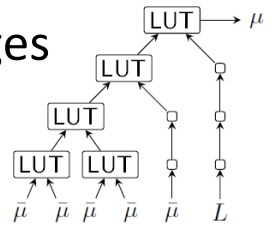
- **Quantization of messages** in fixed-point implementations **has large impact on**
  - Routing, storage, logic **complexity**
  - Decoder **error rate performance** and error floor
- Information theoretic **analysis suggests a non-uniform quantization**
  - Combining with conventional arithmetic is inefficient
- Solution: CNs and VNs based on lookup-tables
  - Information bottleneck analysis provides optimum mapping from input-messages to output messages
  - LUTs optimized for multi-stage as Boolean logic

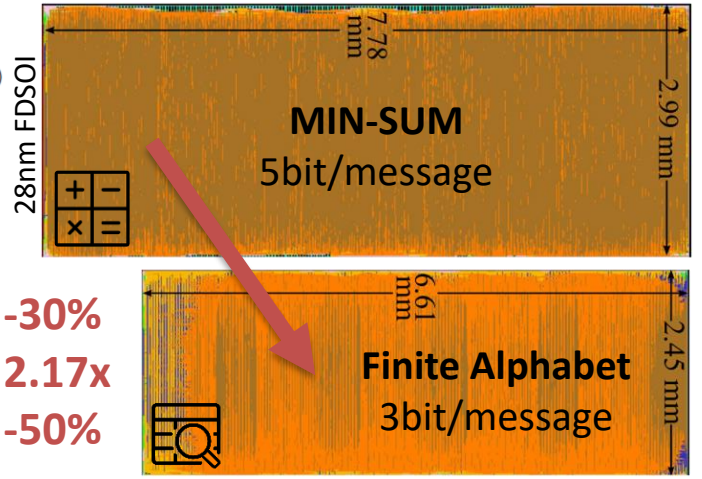


[Ghanaatian, 2017]

$$\sum_{\boldsymbol{x}:\, x_0=\cdots=x_{d_v-1}=x} p_{\mathsf{L}|\mathsf{x}}(L|x_0) \prod_{j=1}^{d_v-1} p^{(i)}_{\overline{\mathsf{m}}|\mathsf{x}}(\bar{\mu}_j|x_j)$$







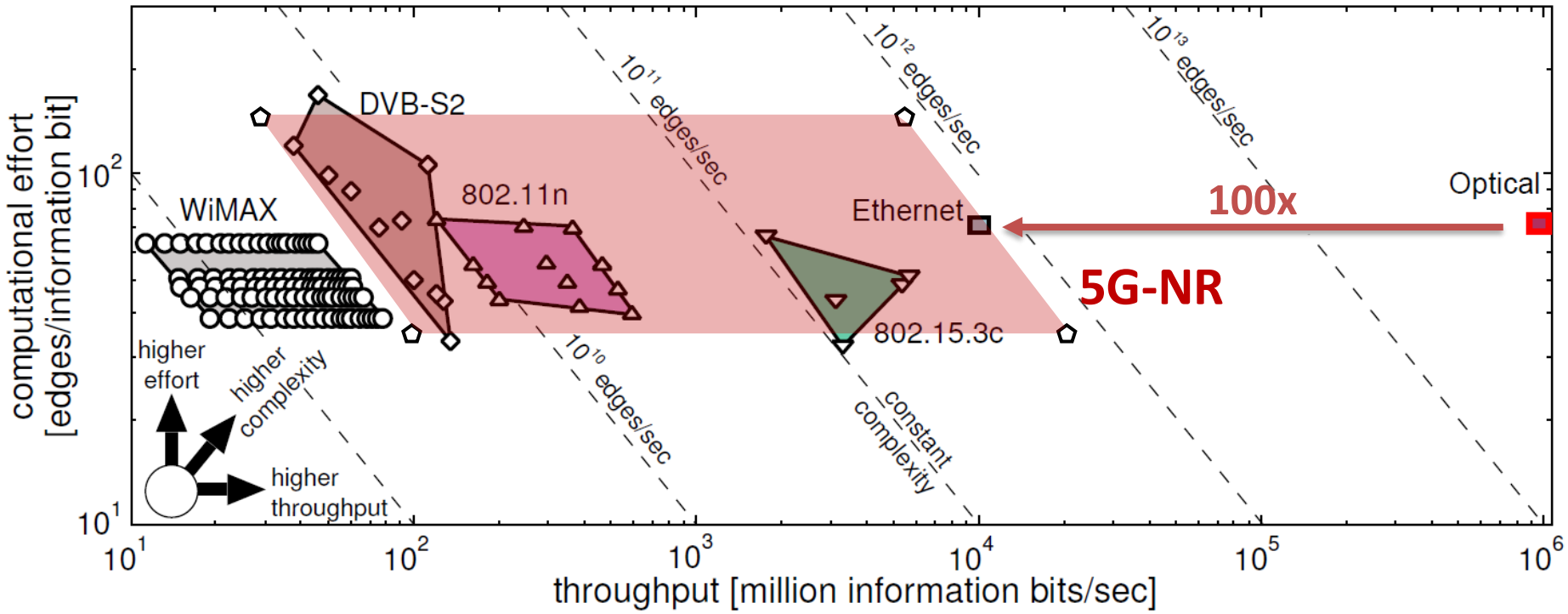
Area	-30%
Throughput	2.17x
Energy/bit	-50%

Throughput >500 Gbps

Ghanaatian, Reza, et al. "A 588-Gb/s LDPC decoder based on finite-alphabet message passing." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26.2 (2017): 329-340.

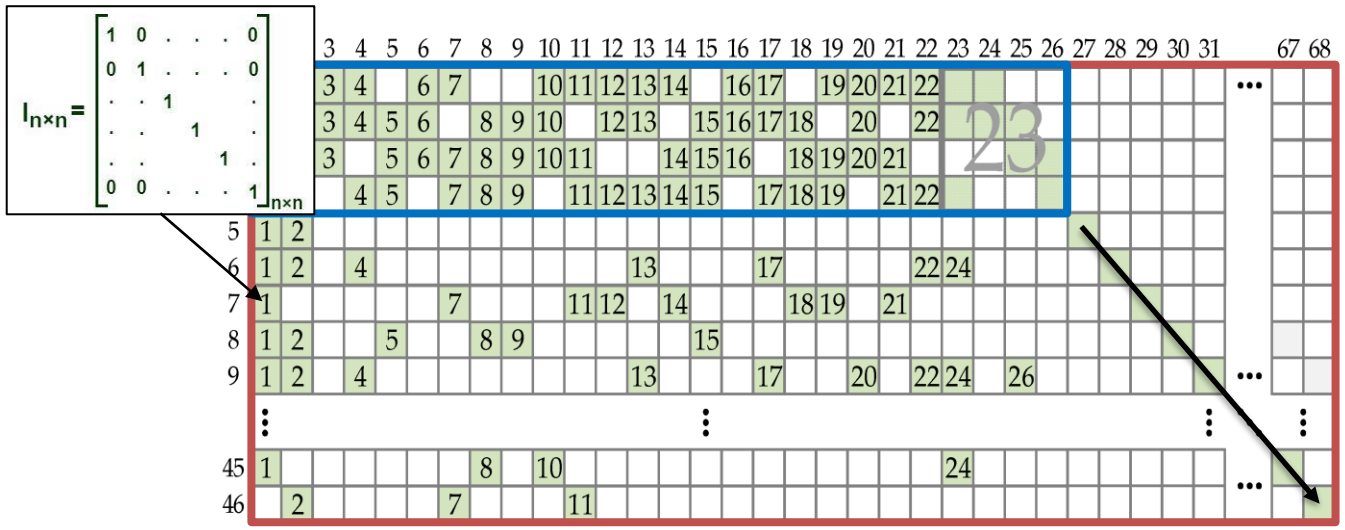
High Throughput for Optical Communications

Computational effort per information bit and **required throughput** determines implementation requirements



Wireless system computational requirements (e.g., 5G) are significantly lower, but require flexibility

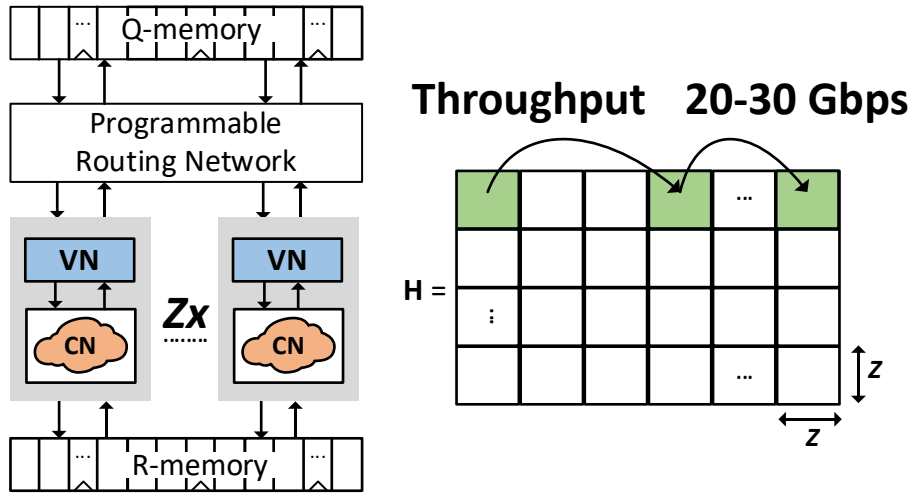
Flexible QC-LDPC Codes and Decoder Architectures for 5G



5G LDPC matrix constructed for flexibility and efficiency

Quasi cyclic code: compact base graph that expands into a larger matrix depending on block size

Raptor-like code: good channels (high rate) require less decoding effort than bad channels



[Ren, 2023]

Decoder architecture optimized for 5G QC-LDPC codes

Layered decoding: row-wise processing of the base graph (good convergence and Zx parallelism)

Block-parallel: decomposes CN and VN calculation into multiple cycles for HW efficiency

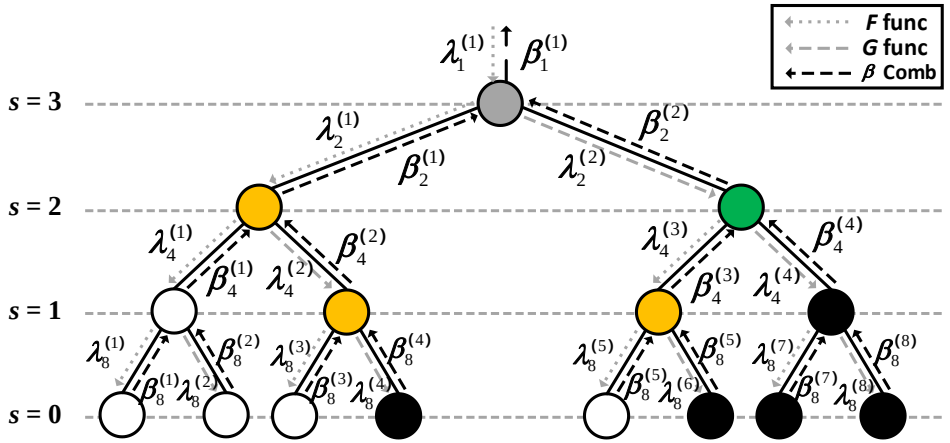
5G-NR Polar Codes for Short Block Lengths and URLLC

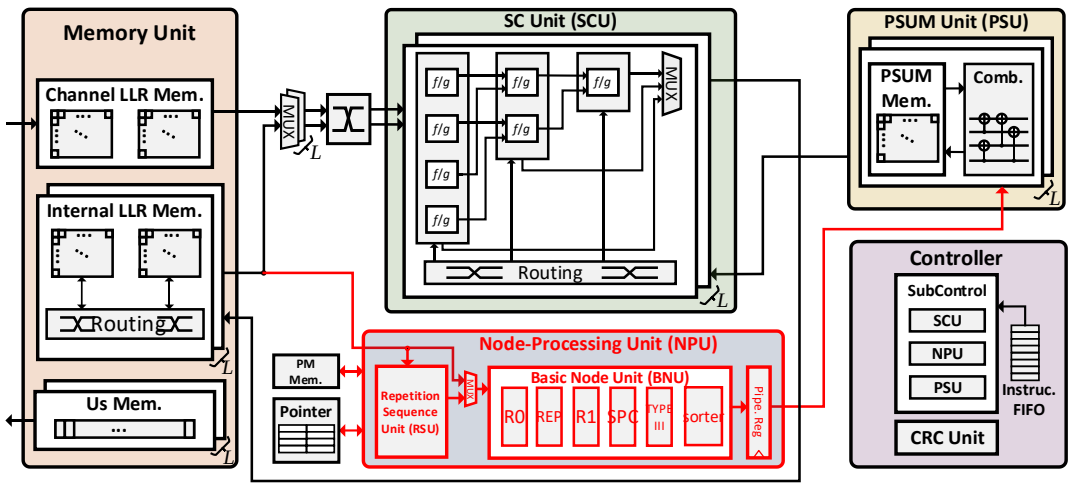
Polar codes: discovered by E. Arikan

- Systematic construction with great flexibility for code length and code rate
- **Good performance for short block lengths** when concatenated with a CRC (for list decoding)
- Used for 5G-NR control channel and considered for URLLC and 6G

Decoding: successive cancellation (SC) algorithm and its list variants for better FER performance

- **Sequential process** following a tree-traversal procedure **decodes bit-by-bit**





[Ren, 2022]

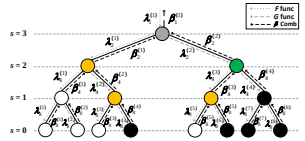
Throughput 3-6 Gbps

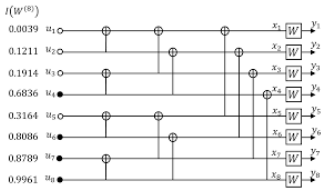
Y. Ren, A. T. Kristensen, Y. Shen, A. Balatsoukas-Stimming, C. Zhang and A. Burg, "A Sequence Repetition Node-Based Successive Cancellation List Decoder for 5G Polar Codes: Algorithm and Implementation," in IEEE Transactions on Signal Processing, 2022

BP Decoding for Everyone

Can we decode Polar codes and other short codes with BP decoding?

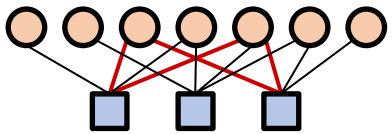
SC List decoding

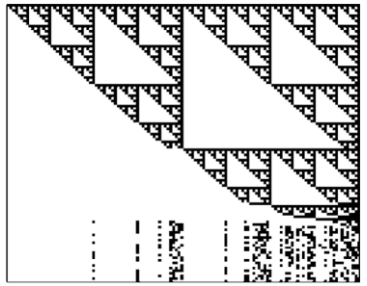




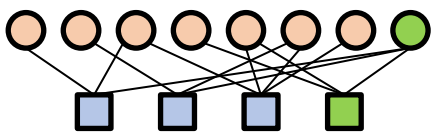
NR Polar Code (128, 20)

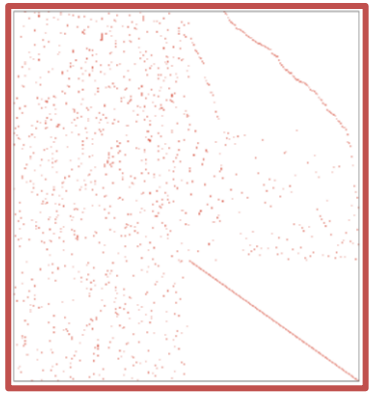
Straightforward Tanner Graph BP





Tanner Graph BP with optimized matrix





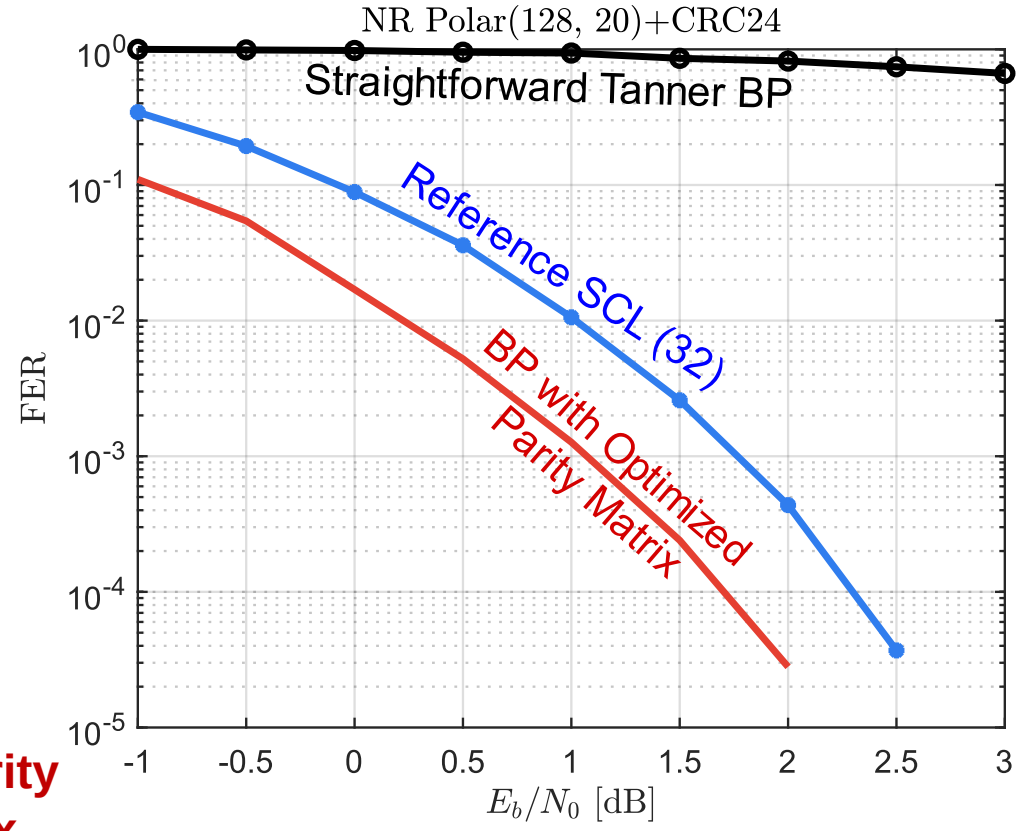
Straightforward derivation of a parity check matrix

↓

BP decoding on a Tanner Graph as for LDPC with high parallelism

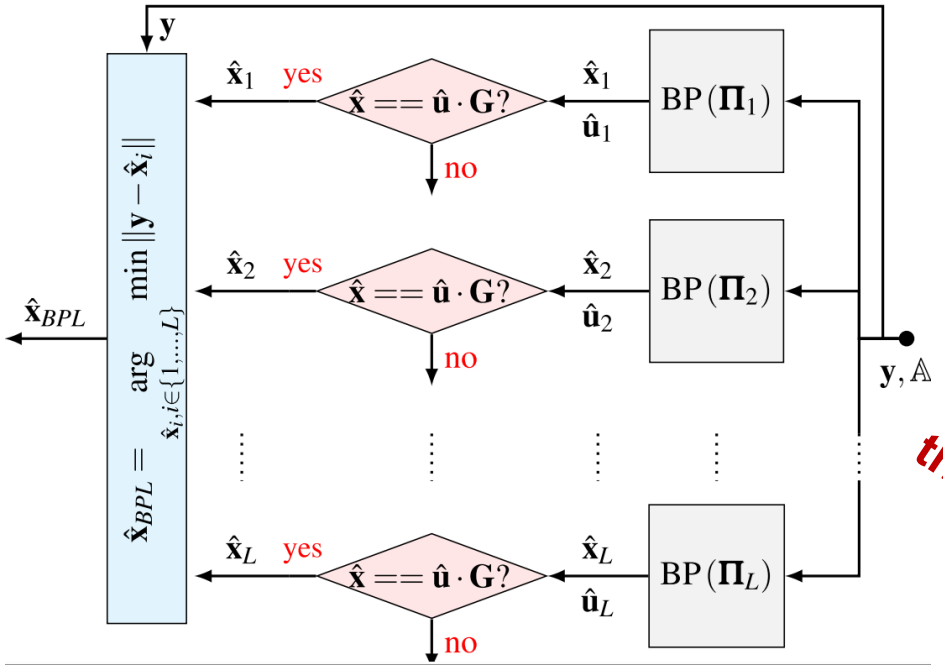
Optimized parity check matrix

Slightly larger, but more sparse with fewer cycles (better BP performance)



Close-to-Optimal Performance with Simple, Suboptimal Decoders

Key insight: a given code can be represented by many different parity check matrices

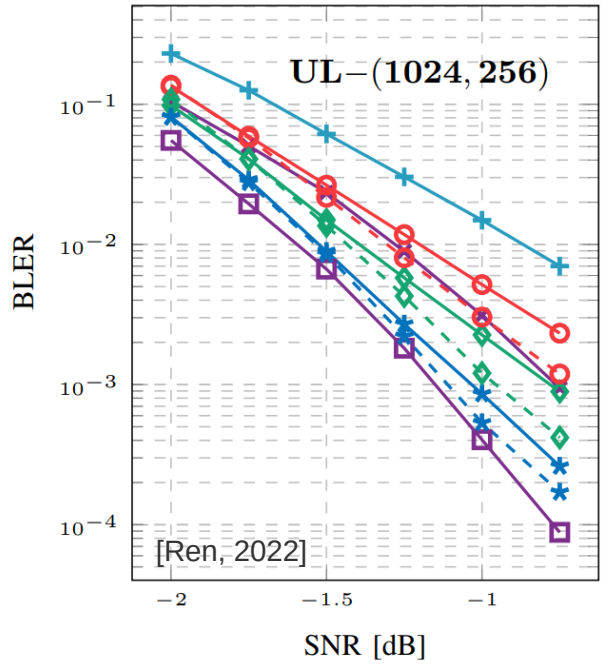


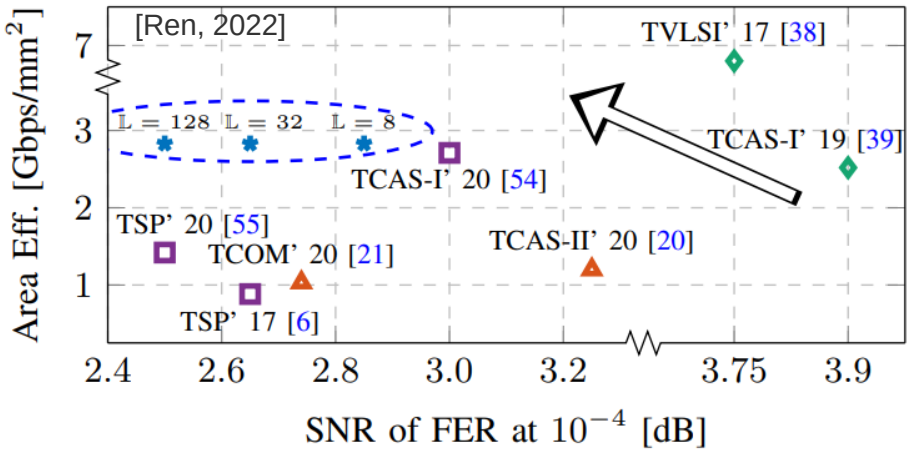
Example: BP List Decoding [Elkelesh, 2018]

Automorphism Ensemble Decoding (AED): multiple sub-optimal decoders try to decode with different parity check matrices

More attempts, better performance

Many sub-optimal decoders more efficient than one optimal decoder

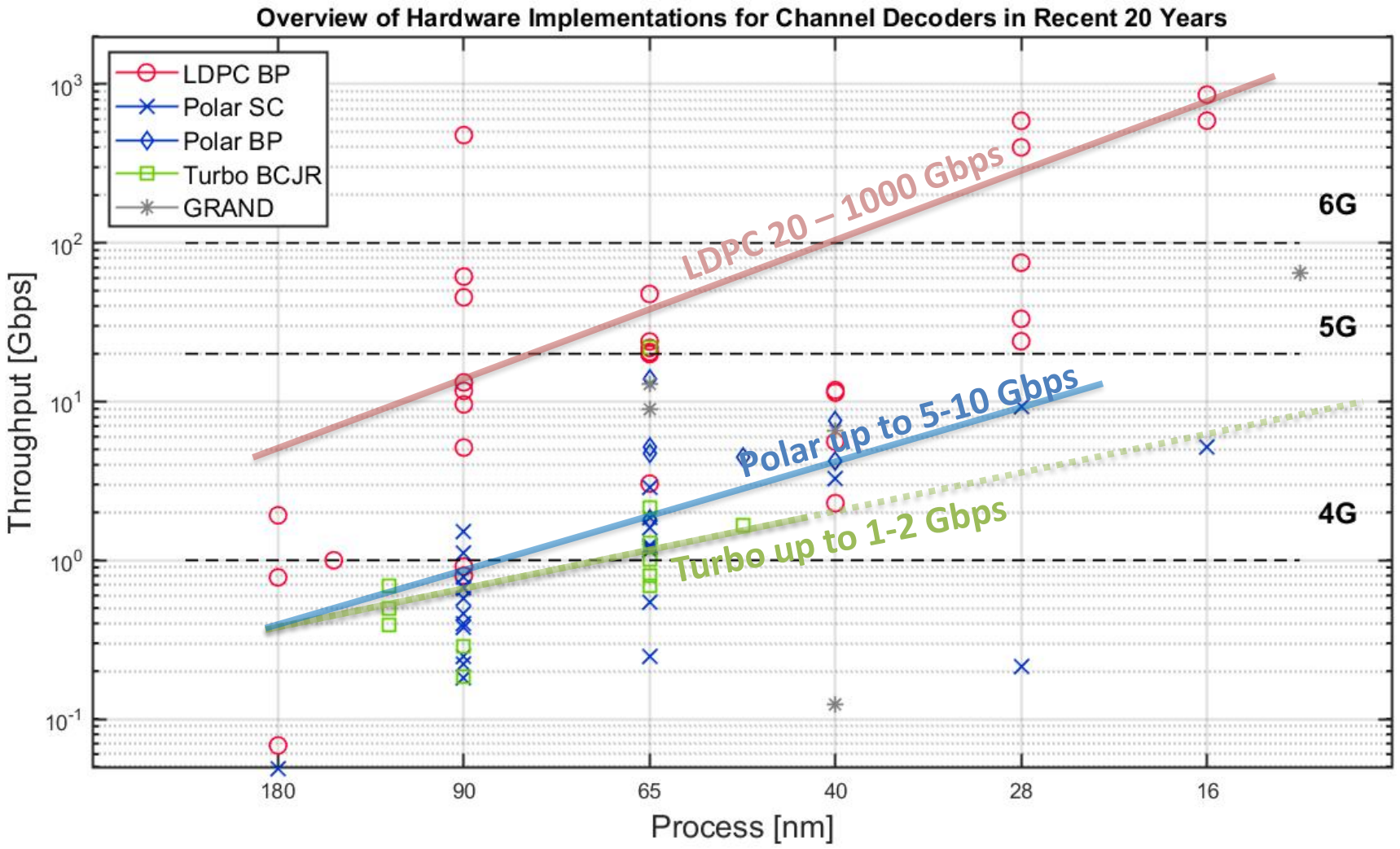




A. Elkelesh, M. Ebada, S. Cammerer and S. ten Brink, "Belief Propagation List Decoding of Polar Codes," in *IEEE Communications Letters*, 2018.

Ren, Yuqing, et al. "High-throughput flexible belief propagation list decoder for polar codes." *arXiv preprint arXiv:2210.13887* (2022).

Conclusions



- Technology continues to scale, but improvements are mostly due to density
- Algorithm choices and optimizations are key to efficient implementation
- Parallel processing is essential to exploit the remains of Moore's law
  - Sequential decoders fall short from scaling
  - BP decoding scales well with technology
- Memory becomes an increasingly serious issue

Acknowledgements

Yuqing Ren

Yifei Shen

Andreas Toftegaard Kristensen

Reza Ghanaatian

Alexios Balatsoukas-Stimming