Polar Codes for Terabit/s Data Rates

Altuğ Süral and Erdal Arıkan

GRC 2019, Bilkent University
Ankara, Turkey
{altug, arikan}@ee.bilkent.edu.tr

March 22th, 2019

Why do we need Tb/s FEC?

- Enable Tb/s wireless infrastructure for contemporary applications

Why do we need Tb/s FEC?

- Enable Tb/s wireless infrastructure for contemporary applications
  - fronthauling/backhauling

Why do we need Tb/s FEC?

- Enable Tb/s wireless infrastructure for contemporary applications
  - fronthauling/backhauling
  - server farming in data centers

Why do we need Tb/s FEC?

- Enable Tb/s wireless infrastructure for contemporary applications
  - fronthauling/backhauling
  - server farming in data centers
  - drone-based communications

Why do we need Tb/s FEC?

- Enable Tb/s wireless infrastructure for contemporary applications
  - fronthauling/backhauling
  - server farming in data centers
  - drone-based communications
  - virtual and augmented reality

Why do we need Tb/s FEC?

- Enable Tb/s wireless infrastructure for contemporary applications
  - fronthauling/backhauling
  - server farming in data centers
  - drone-based communications
  - virtual and augmented reality
  - chip-to-chip and intra-chip communications

Why do we need Tb/s FEC?

- Enable Tb/s wireless infrastructure for contemporary applications
  - fronthauling/backhauling
  - server farming in data centers
  - drone-based communications
  - virtual and augmented reality
  - chip-to-chip and intra-chip communications
  - data kiosks

Why do we need Tb/s FEC?

- Enable Tb/s wireless infrastructure for contemporary applications
  - fronthauling/backhauling
  - server farming in data centers
  - drone-based communications
  - virtual and augmented reality
  - chip-to-chip and intra-chip communications
  - data kiosks
- Contribute to the rapidly emerging standardization studies

Why do we need Tb/s FEC?

- Enable Tb/s wireless infrastructure for contemporary applications
  - fronthauling/backhauling
  - server farming in data centers
  - drone-based communications
  - virtual and augmented reality
  - chip-to-chip and intra-chip communications
  - data kiosks
- Contribute to the rapidly emerging standardization studies
  - IEEE 802.15.3d-2017

Why do we need Tb/s FEC?

- Enable Tb/s wireless infrastructure for contemporary applications
  - fronthauling/backhauling
  - server farming in data centers
  - drone-based communications
  - virtual and augmented reality
  - chip-to-chip and intra-chip communications
  - data kiosks
- Contribute to the rapidly emerging standardization studies
  - IEEE 802.15.3d-2017
  - IEEE 802.15.thz WPAN IG

Why do we need Tb/s FEC?

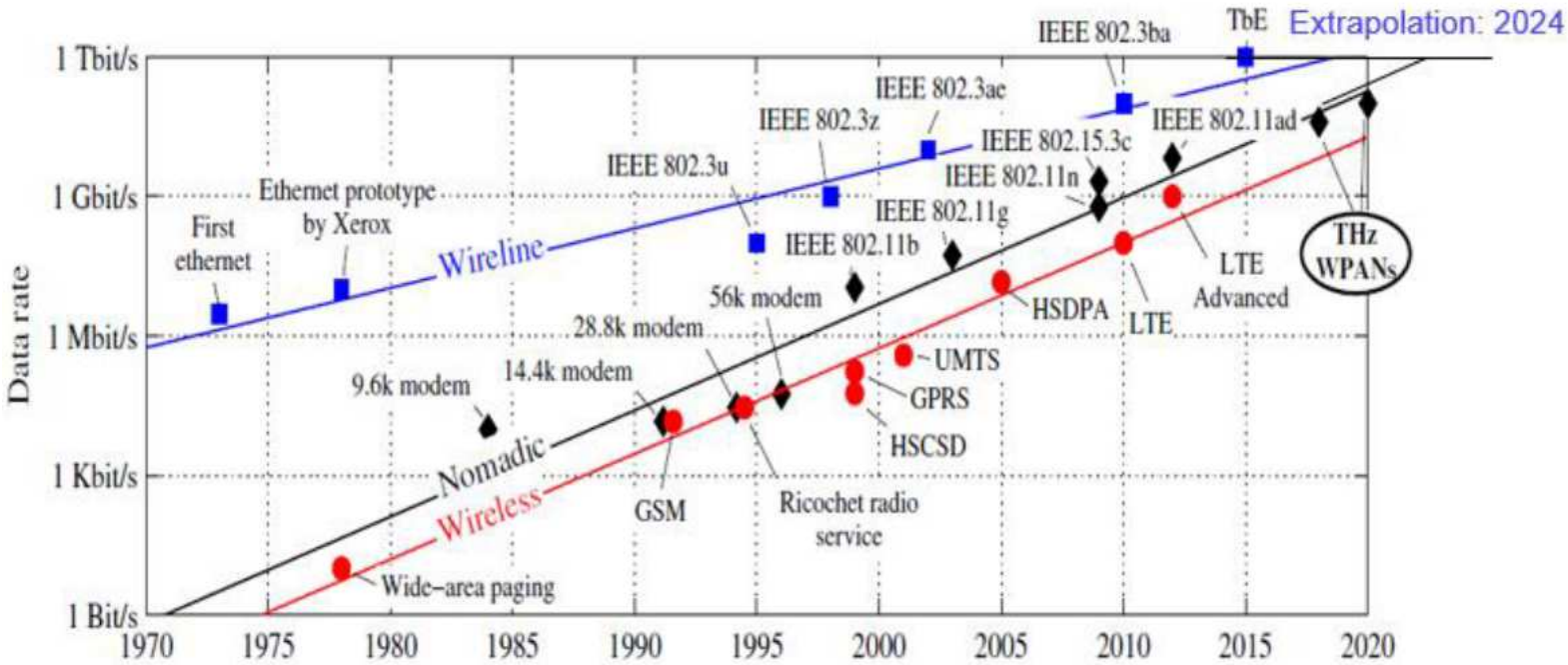
- Enable Tb/s wireless infrastructure for contemporary applications
  - fronthauling/backhauling
  - server farming in data centers
  - drone-based communications
  - virtual and augmented reality
  - chip-to-chip and intra-chip communications
  - data kiosks
- Contribute to the rapidly emerging standardization studies
  - IEEE 802.15.3d-2017
  - IEEE 802.15.thz WPAN IG
  - IEEE 802.11bb Light Communication TG (Li-Fi)

Why do we need Tb/s FEC?

- Enable Tb/s wireless infrastructure for contemporary applications
  - fronthauling/backhauling
  - server farming in data centers
  - drone-based communications
  - virtual and augmented reality
  - chip-to-chip and intra-chip communications
  - data kiosks
- Contribute to the rapidly emerging standardization studies
  - IEEE 802.15.3d-2017
  - IEEE 802.15.thz WPAN IG
  - IEEE 802.11bb Light Communication TG (Li-Fi)
- Provide strong error-resilience for those applications using **polar codes**

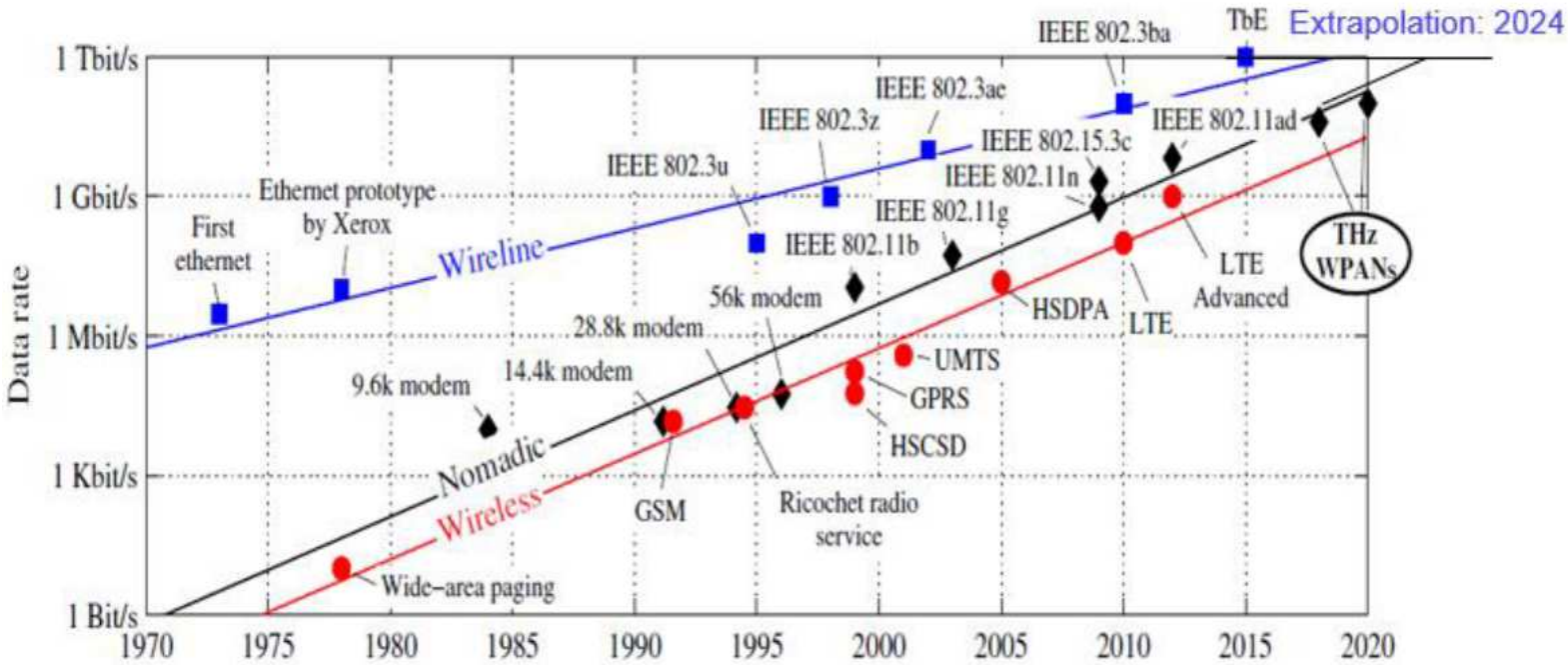
When will we exploit Tb/s FEC?

- The experts in the BRAVE project claim that Tb/s links will become a reality within 4 years [Fet11], [Saa18].



When will we exploit Tb/s FEC?

- The experts in the BRAVE project claim that Tb/s links will become a reality within 4 years [Fet11], [Saa18].

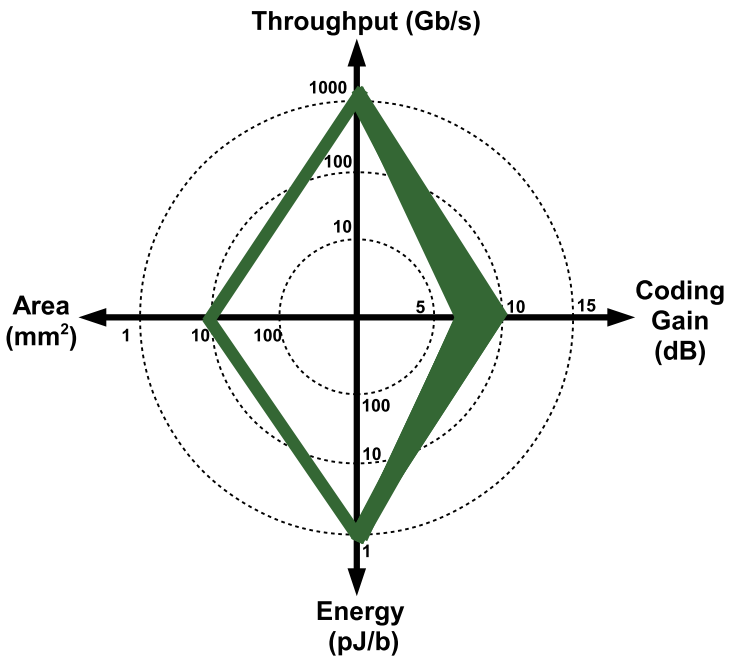


- The 2018 Ethernet Roadmap foresees a demand for Tb/s data rates for 2020 and beyond [Kip18].

What are the requirements of Tb/s FEC?

The ASIC design experts in EPIC project predicts Tb/s FEC throughput within the limits of other important KPIs [Weh17].

Name of KPI	Requirement
Technology	7nm
Throughput	1 Tb/s
Clock freq.	≤ 1 GHz
Core area	≤ 10 mm^2
Power	≤ 1 W
Power den.	≤ 0.1 W/mm^2
Area eff.	≥ 100 Gb/s/mm^2
Energy	≤ 1 pJ/bit



Polar codes for Tb/s FEC

- Develop a specific solution based on polar codes introduced in [Ari09]

Polar codes for Tb/s FEC

- Develop a specific solution based on polar codes introduced in [Ari09]
- Use Majority-Logic aided Successive Cancellation (SC-MJL) decoding algorithm in [Diz17] with several enhancements

Polar codes for Tb/s FEC

- Develop a specific solution based on polar codes introduced in [Ari09]
- Use Majority-Logic aided Successive Cancellation (SC-MJL) decoding algorithm in [Diz17] with several enhancements
  - Wagner decoding for single parity check

Polar codes for Tb/s FEC

- Develop a specific solution based on polar codes introduced in [Ari09]
- Use Majority-Logic aided Successive Cancellation (SC-MJL) decoding algorithm in [Diz17] with several enhancements
  - Wagner decoding for single parity check
  - MAP decoding for repetition

Polar codes for Tb/s FEC

- Develop a specific solution based on polar codes introduced in [Ari09]
- Use Majority-Logic aided Successive Cancellation (SC-MJL) decoding algorithm in [Diz17] with several enhancements
  - Wagner decoding for single parity check
  - MAP decoding for repetition
- Develop a dedicated hardware architecture for the SC-MJL algorithm

Polar codes for Tb/s FEC

- Develop a specific solution based on polar codes introduced in [Ari09]
- Use Majority-Logic aided Successive Cancellation (SC-MJL) decoding algorithm in [Diz17] with several enhancements
  - Wagner decoding for single parity check
  - MAP decoding for repetition
- Develop a dedicated hardware architecture for the SC-MJL algorithm
  - Fully-pipelined processing of the codewords in the form of a set of LLRs

Polar codes for Tb/s FEC

- Develop a specific solution based on polar codes introduced in [Ari09]
- Use Majority-Logic aided Successive Cancellation (SC-MJL) decoding algorithm in [Diz17] with several enhancements
  - Wagner decoding for single parity check
  - MAP decoding for repetition
- Develop a dedicated hardware architecture for the SC-MJL algorithm
  - Fully-pipelined processing of the codewords in the form of a set of LLRs
  - Unrolled modules for each different operation

Polar codes for Tb/s FEC

- Develop a specific solution based on polar codes introduced in [Ari09]
- Use Majority-Logic aided Successive Cancellation (SC-MJL) decoding algorithm in [Diz17] with several enhancements
  - Wagner decoding for single parity check
  - MAP decoding for repetition
- Develop a dedicated hardware architecture for the SC-MJL algorithm
  - Fully-pipelined processing of the codewords in the form of a set of LLRs
  - Unrolled modules for each different operation
- Optimize LLR quantization throughout the decoding

Polar codes for Tb/s FEC

- Develop a specific solution based on polar codes introduced in [Ari09]
- Use Majority-Logic aided Successive Cancellation (SC-MJL) decoding algorithm in [Diz17] with several enhancements
  - Wagner decoding for single parity check
  - MAP decoding for repetition
- Develop a dedicated hardware architecture for the SC-MJL algorithm
  - Fully-pipelined processing of the codewords in the form of a set of LLRs
  - Unrolled modules for each different operation
- Optimize LLR quantization throughout the decoding
- Merge pipelined decoding stages for register balancing/timing

Data flowchart of SC-MJL decoding algorithm

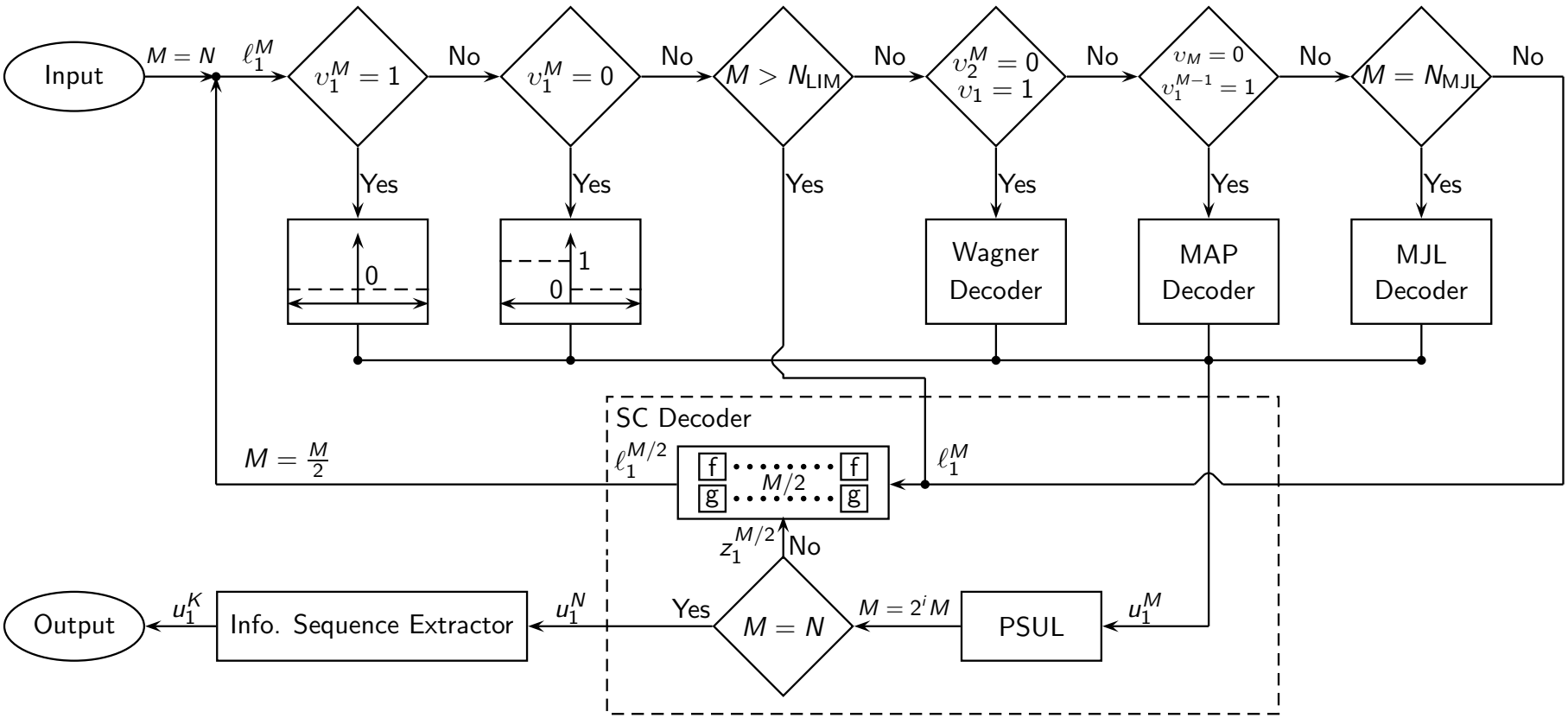
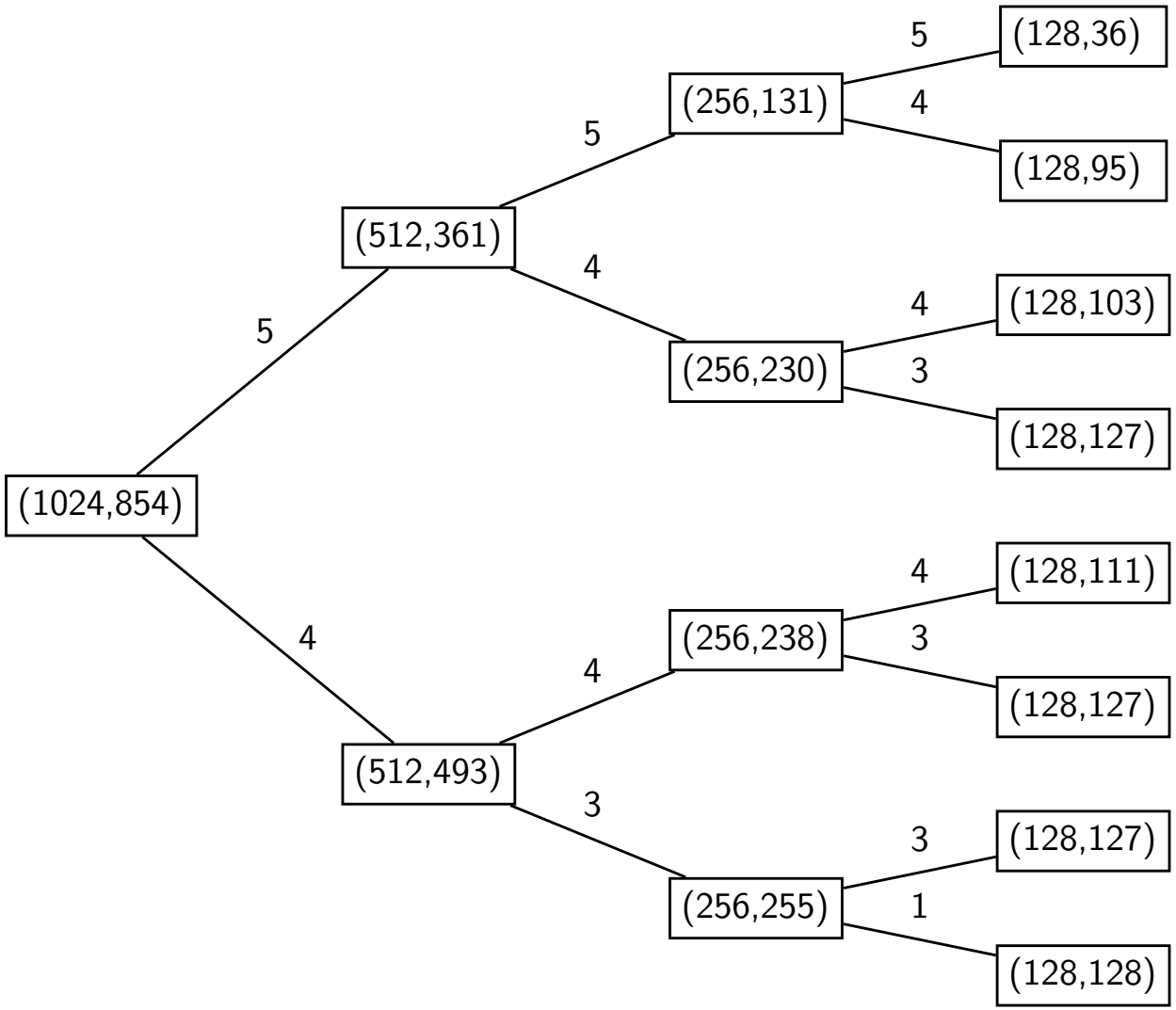


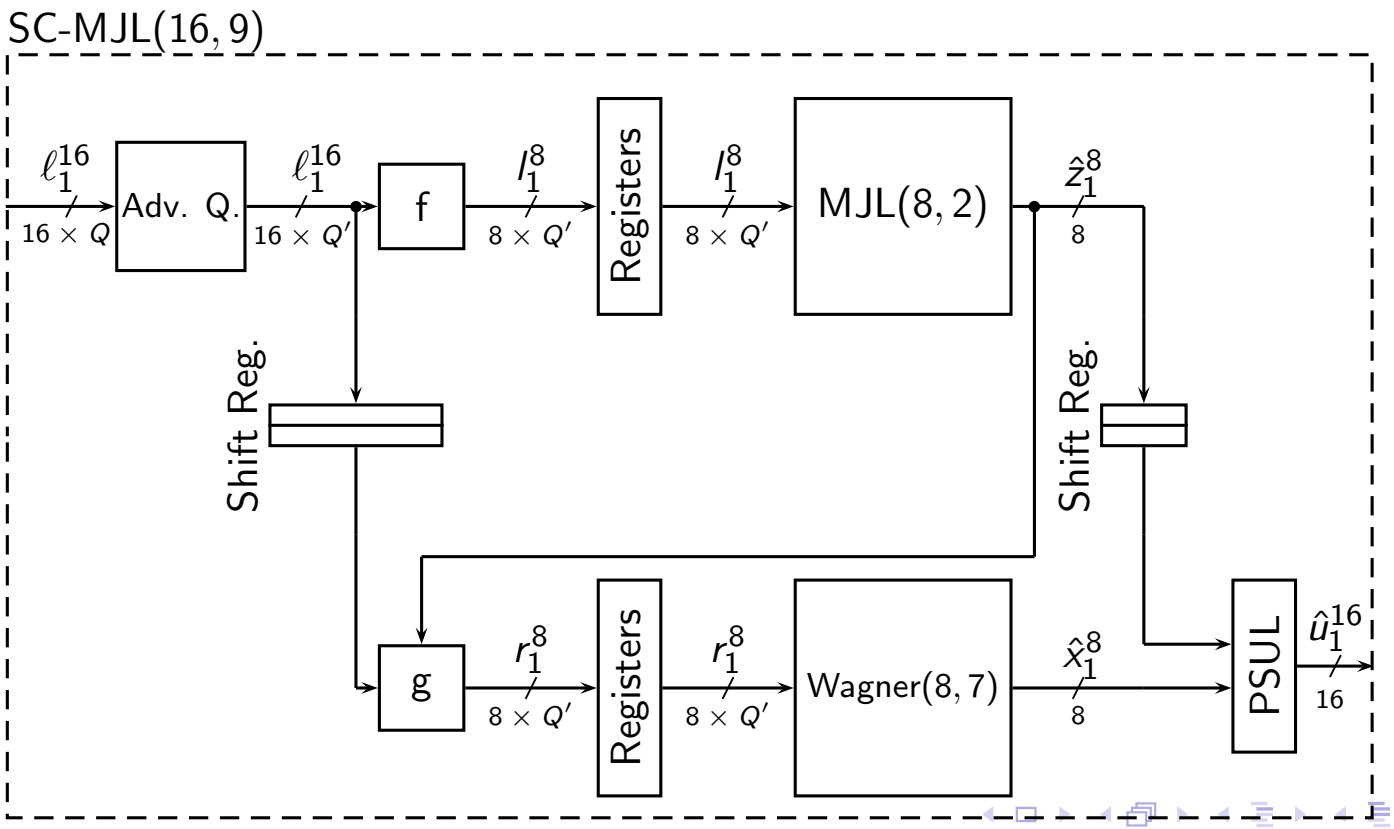
Figure: Data flowchart of SC-MJL decoding algorithm where N is the code block length, K is the number of information bits, v_1^M is the frozen vector with variable constituent block length M, N_{MJL} is the block length of MJL decoder and N_{LIM} is the maximum block length of Wagner and MAP decoders.

Progressive quantization of LLRs inside the decoder



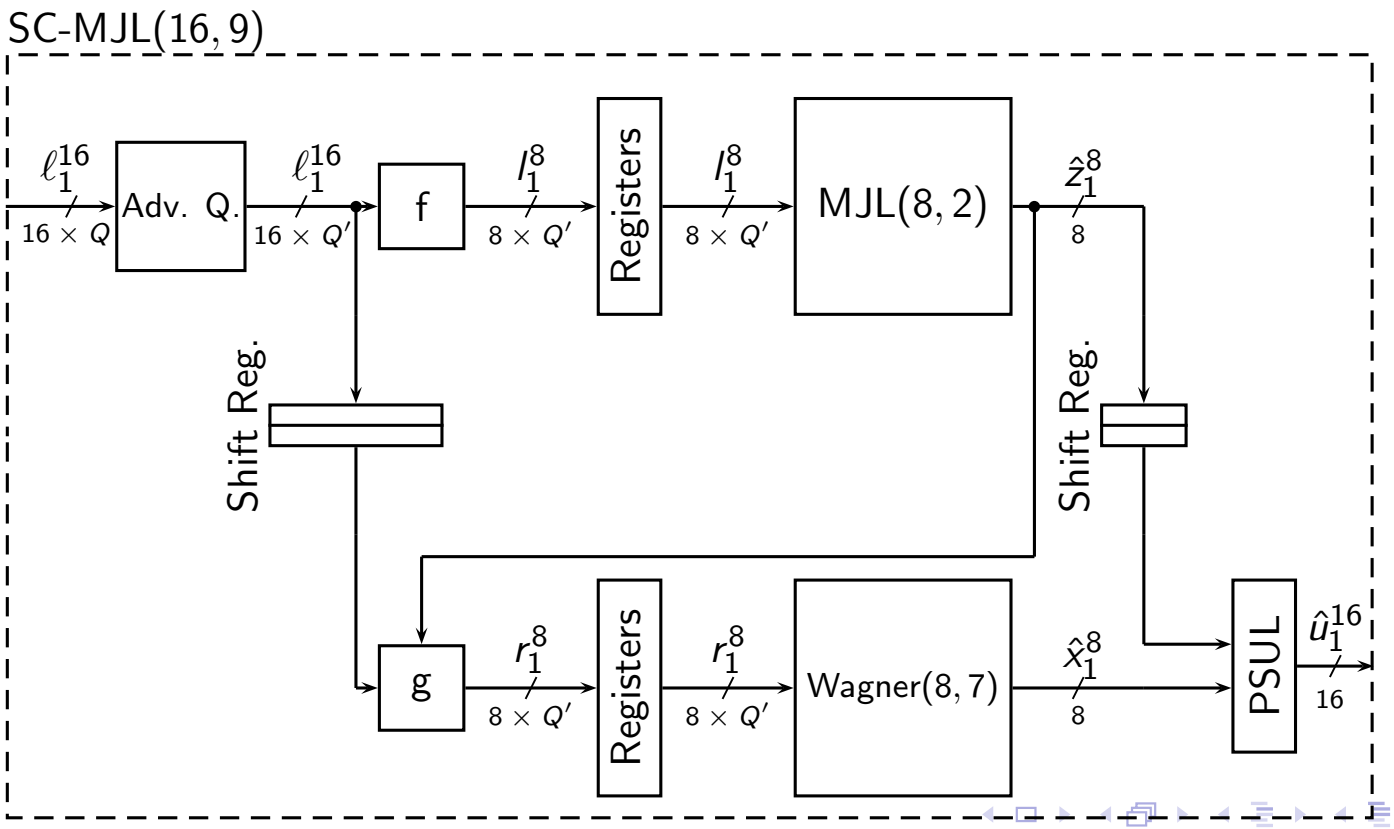
Proposed hardware architecture for SC-MJL

- An example $N = 16$, $K = 9$ design with $N_{\text{MJL}} = 8$ and $N_{\text{LIM}} = 32$ parameters



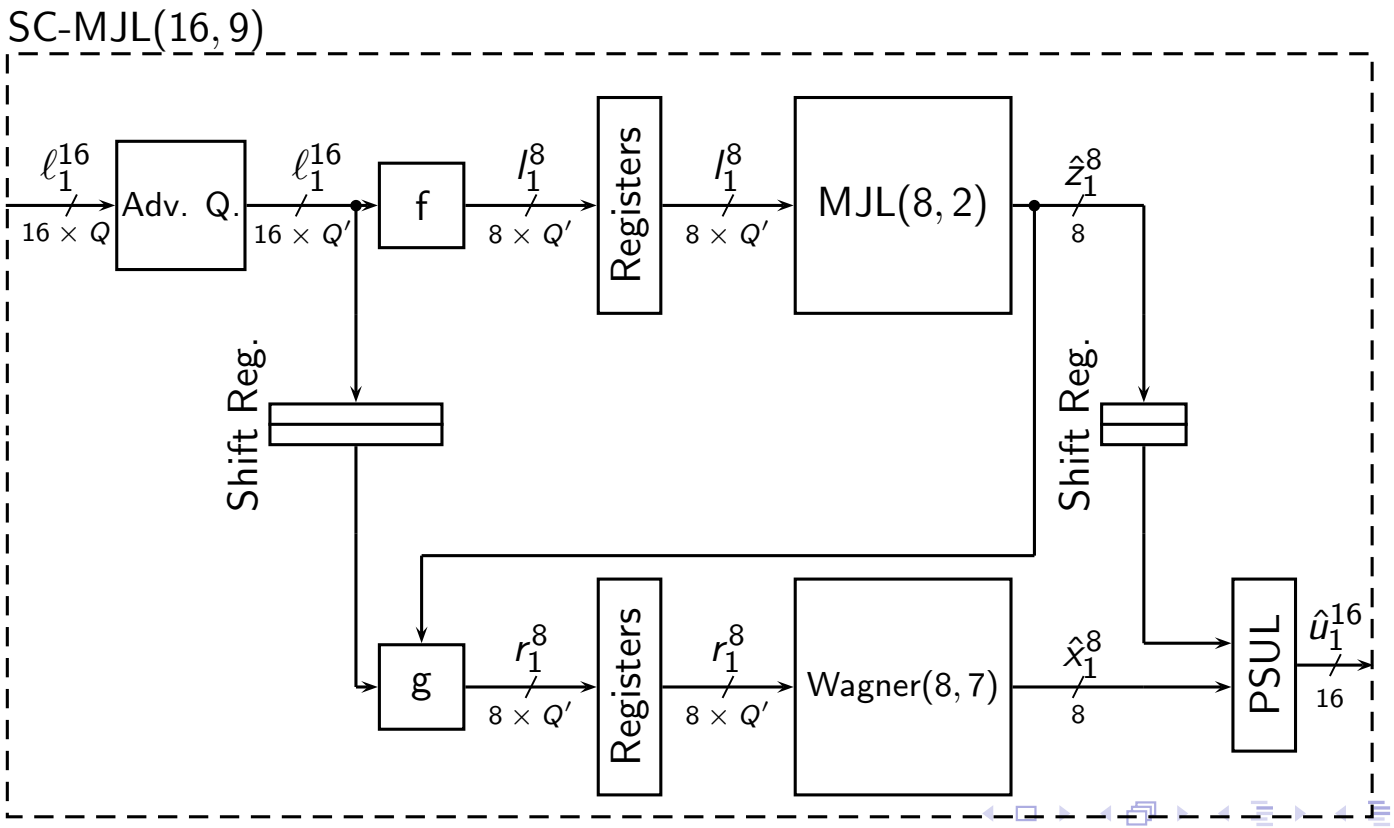
Proposed hardware architecture for SC-MJL

- An example $N = 16$, $K = 9$ design with $N_{\text{MJL}} = 8$ and $N_{\text{LIM}} = 32$ parameters
- Use MJL when $v_1^8 = \{1, 0, 0, 0, 1, 0, 0, 0\}$



Proposed hardware architecture for SC-MJL

- An example $N = 16$, $K = 9$ design with $N_{\text{MJL}} = 8$ and $N_{\text{LIM}} = 32$ parameters
- Use MJL when $v_1^8 = \{1, 0, 0, 0, 1, 0, 0, 0\}$
- Use Wagner when $v_1^8 = \{1, 0, 0, 0, 0, 0, 0, 0\}$



Complexity analysis

- For $n' = \log N_{\mathrm{MJL}}$, the time complexity of the fully-parallel SC-MJL decoder is

$$T_{\mathrm{N}} = 2T_{\mathrm{N}/2} + 2 = \sum_{i=1}^{n-n'} 2^i + 2^{n-n'} = 3\frac{N}{N_{\mathrm{MJL}}} - 2 = \Theta(N).$$

Complexity analysis

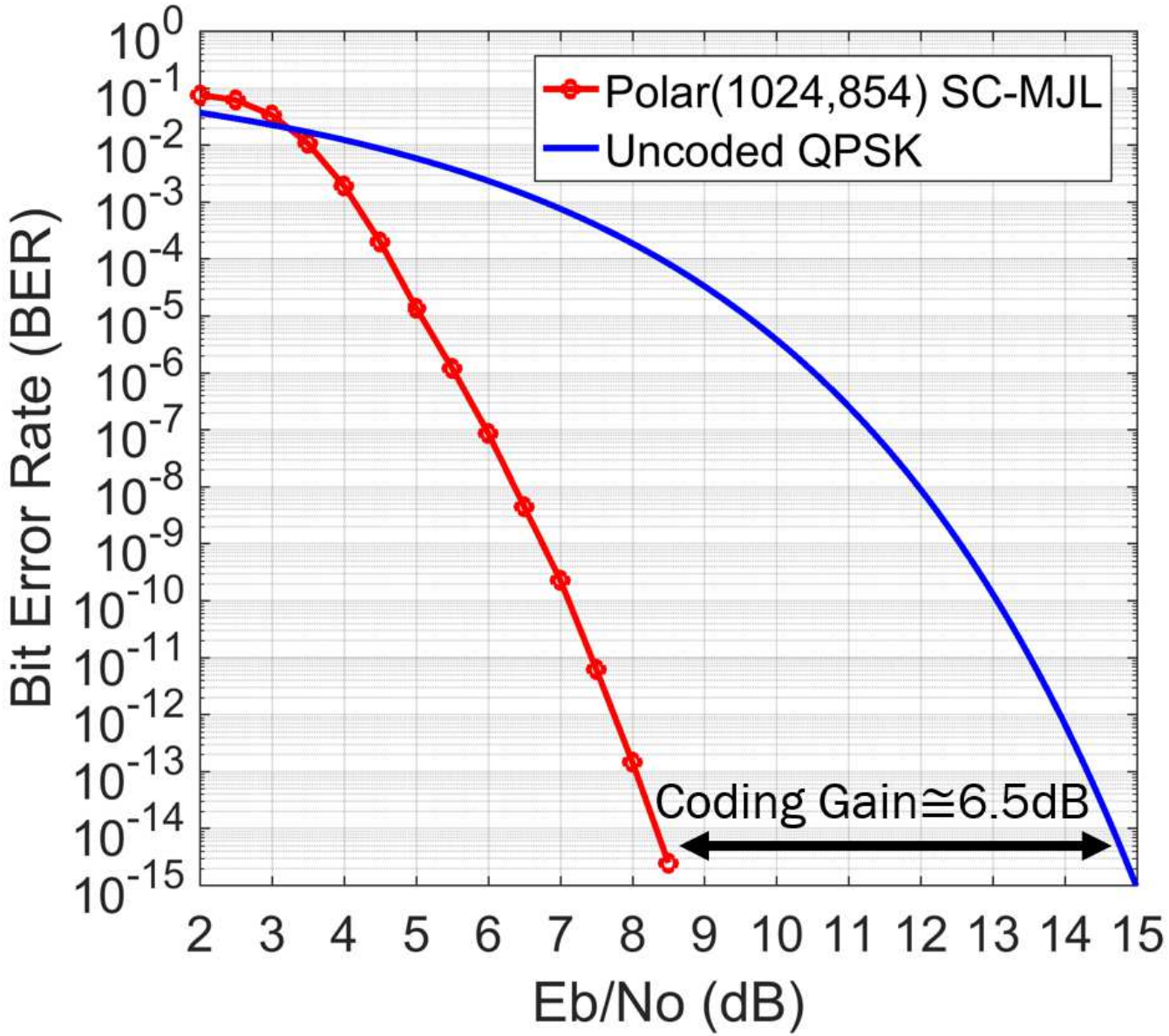
- For $n' = \log N_{\text{MJL}}$, the time complexity of the fully-parallel SC-MJL decoder is

$$T_{\text{N}} = 2T_{\text{N}/2} + 2 = \sum_{i=1}^{n-n'} 2^i + 2^{n-n'} = 3\frac{N}{N_{\text{MJL}}} - 2 = \Theta(N).$$

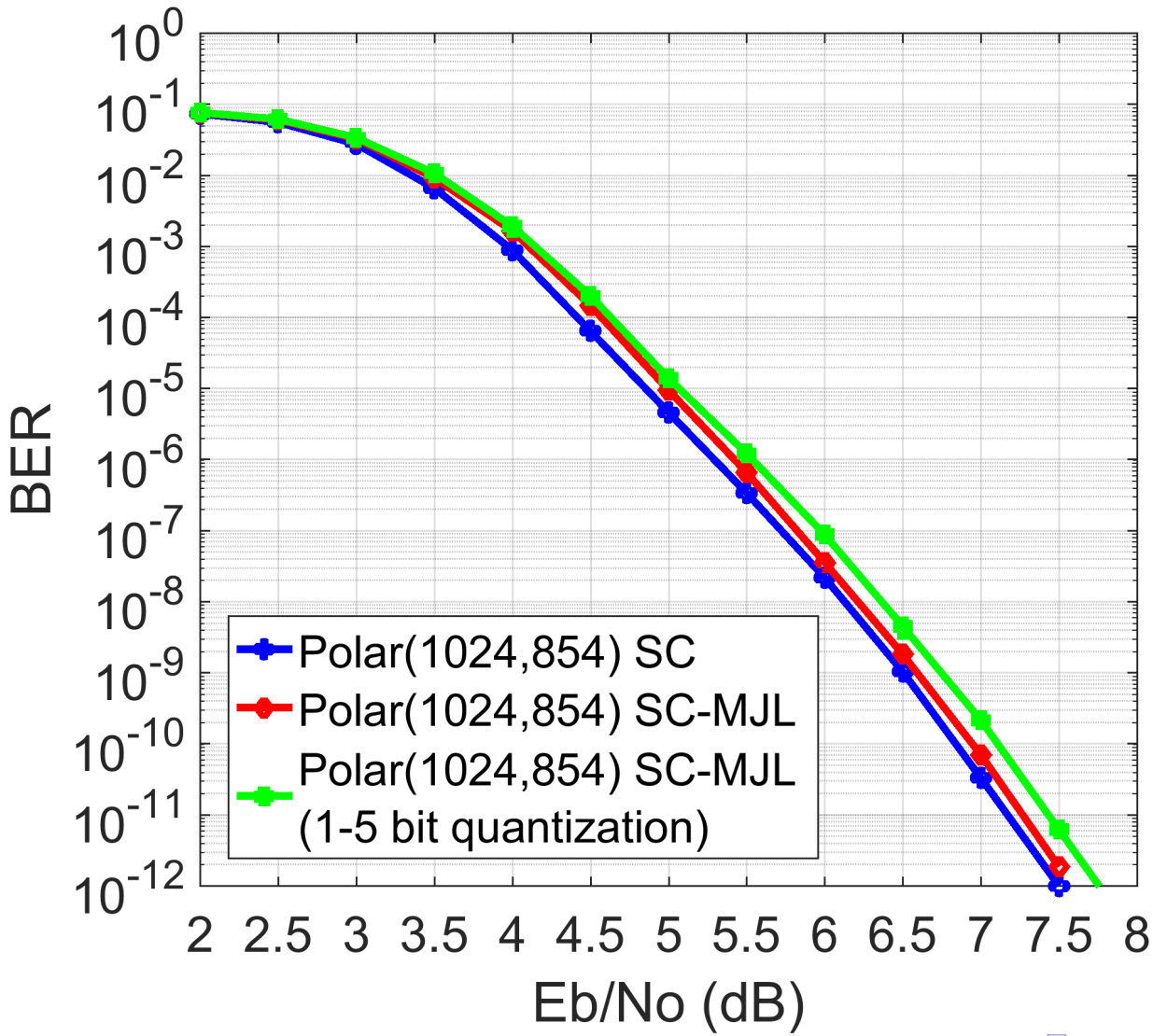
- For $N_{\text{MJL}} \leq N$ and $M_4 = 32Q' + 16$, the memory complexity of the unrolled and fully-pipelined SC-MJL decoder is

$$\begin{aligned} M_{\text{N}} &= 2M_{\text{N}/2} + (Q' + 0.5)\left(1.5\frac{N^2}{N_{\text{MJL}}} - N\right) \\ &= (Q' + 0.5)\left(\frac{1.5}{N_{\text{MJL}}}(2 - 2^{-\log N + \log N_{\text{MJL}}})N^2 - N\log N + 1.5N\right) \\ &= \Theta(N^2 Q'). \end{aligned}$$

FPGA performance of SC-MJL decoder



FPGA performance comparison



ASIC 45nm post-synthesis results of (1024,854) polar code

- Take $N_{\mathrm{MJL}} = 8$ and $N_{\mathrm{LIM}} = 32$

Decoding Algorithm	SC	SC-MJL	SC-MJL	SC-MJL
Quantization (bits)	6	6	5-to-1	5-to-1
Reg. Balancing Arch.	x	x	x	✓
Throughput (Gb/s)			427	
Frequency (MHz)			500	
Latency (μs)	0.31	0.25	0.25	0.08
Latency (Clock cycles)	157	127	127	40
Area (mm^2)	9.8	8.3	6.6	2.4
Area Eff. ($\mathrm{Gb/s/mm}^2$)	43.5	51.4	65.0	175.2
Power (W)	4.6	3.1	2.3	1.0
Pow. Den. ($\mathrm{W/mm}^2$)	0.47	0.38	0.36	0.42
Energy (pJ/bit)	10.9	7.3	5.5	2.4

Scaled 7nm results in ASIC

- Use ITRS CMOS scaling rules in [Gra15] from 45nm to 7nm

Decoding Algorithm	SC	SC-MJL	SC-MJL	SC-MJL
Quantization (bits)	6	6	5-to-1	5-to-1
Reg. Balancing Arch.	x	x	x	✓
Throughput (Gb/s)			1000	
Frequency (MHz)			585.5	
Area (mm^2)			10	
Area Eff. ($Gb/s/mm^2$)			100	
Power (W)	1.69	1.14	0.85	0.37
Pow. Den. (W/mm^2)	0.17	0.11	0.09	0.04
Energy Eff. (pJ/bit)	1.69	1.14	0.85	0.37
Latency (μs)	0.27	0.22	0.22	0.07
Latency (Clock Cycles)	157	127	127	40

Scaled 7nm results in ASIC

- Use ITRS CMOS scaling rules in [Gra15] from 45nm to 7nm
- Use two identical spatially parallel polar decoders

Decoding Algorithm	SC	SC-MJL	SC-MJL	SC-MJL
Quantization (bits)	6	6	5-to-1	5-to-1
Reg. Balancing Arch.	x	x	x	✓
Throughput (Gb/s)			1000	
Frequency (MHz)			585.5	
Area (mm^2)			10	
Area Eff. ($Gb/s/mm^2$)			100	
Power (W)	1.69	1.14	0.85	0.37
Pow. Den. (W/mm^2)	0.17	0.11	0.09	0.04
Energy Eff. (pJ/bit)	1.69	1.14	0.85	0.37
Latency (μs)	0.27	0.22	0.22	0.07
Latency (Clock Cycles)	157	127	127	40

Scaled 7nm results in ASIC

- Use ITRS CMOS scaling rules in [Gra15] from 45nm to 7nm
- Use two identical spatially parallel polar decoders
- Multiply frequency by 0.3

Decoding Algorithm	SC	SC-MJL	SC-MJL	SC-MJL
Quantization (bits)	6	6	5-to-1	5-to-1
Reg. Balancing Arch.	x	x	x	✓
Throughput (Gb/s)			1000	
Frequency (MHz)			585.5	
Area (mm^2)			10	
Area Eff. ($Gb/s/mm^2$)			100	
Power (W)	1.69	1.14	0.85	0.37
Pow. Den. (W/mm^2)	0.17	0.11	0.09	0.04
Energy Eff. (pJ/bit)	1.69	1.14	0.85	0.37
Latency (μs)	0.27	0.22	0.22	0.07
Latency (Clock Cycles)	157	127	127	40

Summary

- 1 Tb/s FEC appears feasible with a 6.5 dB coding gain

Summary

- 1 Tb/s FEC appears feasible with a 6.5 dB coding gain
- The proposed solution brought together some existing techniques

Summary

- 1 Tb/s FEC appears feasible with a 6.5 dB coding gain
- The proposed solution brought together some existing techniques
  - SC decoder in initial stages of decoding where parallelism is high

Summary

- 1 Tb/s FEC appears feasible with a 6.5 dB coding gain
- The proposed solution brought together some existing techniques
  - SC decoder in initial stages of decoding where parallelism is high
  - MJL decoding for speeding up decisions

Summary

- 1 Tb/s FEC appears feasible with a 6.5 dB coding gain
- The proposed solution brought together some existing techniques
  - SC decoder in initial stages of decoding where parallelism is high
  - MJL decoding for speeding up decisions
  - Progressive quantization to reduce memory usage

Summary

- 1 Tb/s FEC appears feasible with a 6.5 dB coding gain
- The proposed solution brought together some existing techniques
  - SC decoder in initial stages of decoding where parallelism is high
  - MJL decoding for speeding up decisions
  - Progressive quantization to reduce memory usage
  - Register balancing to reducing pipeline depth by combining simple steps

Summary

- 1 Tb/s FEC appears feasible with a 6.5 dB coding gain
- The proposed solution brought together some existing techniques
  - SC decoder in initial stages of decoding where parallelism is high
  - MJL decoding for speeding up decisions
  - Progressive quantization to reduce memory usage
  - Register balancing to reducing pipeline depth by combining simple steps
- Storage complexity dominates the design

Summary

- 1 Tb/s FEC appears feasible with a 6.5 dB coding gain
- The proposed solution brought together some existing techniques
  - SC decoder in initial stages of decoding where parallelism is high
  - MJL decoding for speeding up decisions
  - Progressive quantization to reduce memory usage
  - Register balancing to reducing pipeline depth by combining simple steps
- Storage complexity dominates the design
- Future studies to improve the KPIs

Summary

- 1 Tb/s FEC appears feasible with a 6.5 dB coding gain
- The proposed solution brought together some existing techniques
  - SC decoder in initial stages of decoding where parallelism is high
  - MJL decoding for speeding up decisions
  - Progressive quantization to reduce memory usage
  - Register balancing to reducing pipeline depth by combining simple steps
- Storage complexity dominates the design
- Future studies to improve the KPIs
  - Complete ASIC back-end design flow

Summary

- 1 Tb/s FEC appears feasible with a 6.5 dB coding gain
- The proposed solution brought together some existing techniques
  - SC decoder in initial stages of decoding where parallelism is high
  - MJL decoding for speeding up decisions
  - Progressive quantization to reduce memory usage
  - Register balancing to reducing pipeline depth by combining simple steps
- Storage complexity dominates the design
- Future studies to improve the KPIs
  - Complete ASIC back-end design flow
  - Implement SC-List to enhance the coding gain up to 10 dB

Summary

- 1 Tb/s FEC appears feasible with a 6.5 dB coding gain
- The proposed solution brought together some existing techniques
  - SC decoder in initial stages of decoding where parallelism is high
  - MJL decoding for speeding up decisions
  - Progressive quantization to reduce memory usage
  - Register balancing to reducing pipeline depth by combining simple steps
- Storage complexity dominates the design
- Future studies to improve the KPIs
  - Complete ASIC back-end design flow
  - Implement SC-List to enhance the coding gain up to 10 dB
  - Find solutions for flexibility in terms of N and K

Acknowledgements

- We thank Prof. Orhan Arıkan for the valuable discussions and ideas about the progressive quantization method.
- We thank Goksu Sezer for VHDL implementation of the SC-MJL algorithm and Yiğit Ertuğrul for carring out the simulations.
- We thank Prof. Abdullah Atalar for his effort to bring TSMC 40nm CMOS low-power and standard cell libraries to Bilkent University.
- This work has been carried out by a support from the EPIC project, with the funding from the European Union's Horizon 2020 research and innovation programme under grant No. 760150.

References

Erdal Arikan.
Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels.
IEEE Transactions on Information Theory, 55(7):3051–3073, July 2009.

Onur Dizdar.
High Throughput Decoding Methods and Architectures for Polar Codes with High Energy-Efficiency and Low Latency.
PhD thesis, Bilkent University, 2017.

Gehard Fettweis.
Entering the path towards terabit/s wireless links.
pages 1 – 6, 04 2011.

Mart Graef.
2015 International Technology Roadmap for Semiconductors (ITRS) Section 5: More Moore.
Technical report, 2015.

Scott Kipp.
EthernetRoadmap-2018-side1-1600x1200.jpg (JPEG image, 1600x120 pixels) - scaled (76%), 2018.

Majed Saad.
Beyond-5G Wireless Tbps Scenarios and Requirements, 2018.

Norbert Wehn.
EPIC - Enabling practical wireless Tb/s communications with next generation channel coding.
https://epic-h2020.eu/results, 2017.

Thank you!