Polar Codes for Terabit/s Data Rates

Erdal Arıkan, Altuğ Süral, E. Göksu Sezer

Bilkent University and Polaran Ltd.
Ankara, Turkey

Dec. 4th, 2018
International Symposium on Turbo Codes and Iterative Information Processing 2018 (ISTC 2018)
Hong Kong

Goals

- Provide some motivation and background for the problem

Goals

- Provide some motivation and background for the problem
- Discuss challenges for Tb/s Forward Error Correction (FEC) with current VLSI technology

Goals

- Provide some motivation and background for the problem
- Discuss challenges for Tb/s Forward Error Correction (FEC) with current VLSI technology
- Present a solution based on polar codes developed jointly in a H2020 project (EPIC)

Goals

- Provide some motivation and background for the problem
- Discuss challenges for Tb/s Forward Error Correction (FEC) with current VLSI technology
- Present a solution based on polar codes developed jointly in a H2020 project (EPIC)
- State some remaining challenges

Why do we need Tb/s FEC?

- Exploit the vast spectrum potential above 90 GHz

Why do we need Tb/s FEC?

- Exploit the vast spectrum potential above 90 GHz
- Bring wireless systems to optical speeds for new applications for use in

Why do we need Tb/s FEC?

- Exploit the vast spectrum potential above 90 GHz
- Bring wireless systems to optical speeds for new applications for use in
  - fronthauling/backhauling

Why do we need Tb/s FEC?

- Exploit the vast spectrum potential above 90 GHz
- Bring wireless systems to optical speeds for new applications for use in
  - fronthauling/backhauling
  - data centers

Why do we need Tb/s FEC?

- Exploit the vast spectrum potential above 90 GHz
- Bring wireless systems to optical speeds for new applications for use in
  - fronthauling/backhauling
  - data centers
  - on/intra chip communications

Why do we need Tb/s FEC?

- Exploit the vast spectrum potential above 90 GHz
- Bring wireless systems to optical speeds for new applications for use in
  - fronthauling/backhauling
  - data centers
  - on/intra chip communications
  - data kiosks

Why do we need Tb/s FEC?

- Exploit the vast spectrum potential above 90 GHz
- Bring wireless systems to optical speeds for new applications for use in
  - fronthauling/backhauling
  - data centers
  - on/intra chip communications
  - data kiosks
  - ...

Why do we need Tb/s FEC?

- Exploit the vast spectrum potential above 90 GHz
- Bring wireless systems to optical speeds for new applications for use in
  - fronthauling/backhauling
  - data centers
  - on/intra chip communications
  - data kiosks
  - ...
- Relevant standardization already underway (IEEE 802.15.3d, LiFi)

Why do we need Tb/s FEC?

- Exploit the vast spectrum potential above 90 GHz
- Bring wireless systems to optical speeds for new applications for use in
  - fronthauling/backhauling
  - data centers
  - on/intra chip communications
  - data kiosks
  - ...
- Relevant standardization already underway (IEEE 802.15.3d, LiFi)
- Other technologies are also needed for Tb/s communications but FEC is one of the most complex part of the transmission chain

What are the challenges?

- Channel coding requirements not extraordinary

What are the challenges?

- Channel coding requirements not extraordinary
  - Reliability: frame error rate of 10^{-6} to 10^{-12} without retransmissions

What are the challenges?

- Channel coding requirements not extraordinary
  - Reliability: frame error rate of 10^{-6} to 10^{-12} without retransmissions
- Implementation challenges are beyond the state-of-the-art

What are the challenges?

- Channel coding requirements not extraordinary
  - Reliability: frame error rate of 10^{-6} to 10^{-12} without retransmissions
- Implementation challenges are beyond the state-of-the-art
- We will use the targets set by the EPIC Project experts

Technology	7nm
Throughput	1 Tb/s
Clock freq.	$\leq$ 1 GHz
Silicon area	$\leq$ 10 mm^2
Pow. Den.	$\leq$ 0.1 W/mm^2
Area Eff.	$\geq$ 100 $Gb/s/mm^2$
Energy Eff.	$\leq$ 1 pJ/bit

What are the challenges?

- Channel coding requirements not extraordinary
  - Reliability: frame error rate of 10^{-6} to 10^{-12} without retransmissions
- Implementation challenges are beyond the state-of-the-art
- We will use the targets set by the EPIC Project experts

Technology	7nm
Throughput	1 Tb/s
Clock freq.	$\leq$ 1 GHz
Silicon area	$\leq$ 10 mm^2
Pow. Den.	$\leq$ 0.1 W/mm^2
Area Eff.	$\geq$ 100 $Gb/s/mm^2$
Energy Eff.	$\leq$ 1 pJ/bit

- Goal: Obtain the best coding gain per code family subject to these constraints

Feasibility check

- Does uncoded transmission meet the targets?

Feasibility check

- Does uncoded transmission meet the targets?
- Uncoded transmission (design at 40nm, scaled to 7nm on paper)

	EPIC target	Uncoded
Technology (nm)	7	7
Throughput (Tb/s)	1	1
Clock freq. (GHz)	≤ 1	1
Silicon area (mm^2)	≤ 10	10
Pow. Den. (W/mm^2)	≤ 0.1	2.3×10^{-4}
Area Eff. ($Gb/s/mm^2$)	≥ 100	100
Energy Eff. (pJ/bit)	≤ 1	2.3×10^{-3}

Feasibility check

- Does uncoded transmission meet the targets?
- Uncoded transmission (design at 40nm, scaled to 7nm on paper)

	EPIC target	Uncoded
Technology (nm)	7	7
Throughput (Tb/s)	1	1
Clock freq. (GHz)	≤ 1	1
Silicon area (mm^2)	≤ 10	10
Pow. Den. (W/mm^2)	≤ 0.1	2.3×10^{-4}
Area Eff. ($Gb/s/mm^2$)	≥ 100	100
Energy Eff. (pJ/bit)	≤ 1	2.3×10^{-3}

- There is room for $1000/2.3 = 435$ times more complex decoding operations relative to uncoded

Feasibility check

- Does uncoded transmission meet the targets?
- Uncoded transmission (design at 40nm, scaled to 7nm on paper)

	EPIC target	Uncoded
Technology (nm)	7	7
Throughput (Tb/s)	1	1
Clock freq. (GHz)	≤ 1	1
Silicon area (mm^2)	≤ 10	10
Pow. Den. (W/mm^2)	≤ 0.1	2.3×10^{-4}
Area Eff. ($Gb/s/mm^2$)	≥ 100	100
Energy Eff. (pJ/bit)	≤ 1	2.3×10^{-3}

- There is room for $1000/2.3 = 435$ times more complex decoding operations relative to uncoded
- Design space is narrowed significantly but still interesting

I/O Bottleneck

- Decoder input bus width

$$W = \frac{\gamma \times Q}{R \times f_c}$$

where

I/O Bottleneck

- Decoder input bus width

$$W = \frac{\gamma \times Q}{R \times f_c}$$

where

  - γ is the throughput in b/s,

I/O Bottleneck

- Decoder input bus width

$$W = \frac{\gamma \times Q}{R \times f_c}$$

where

  - γ is the throughput in b/s,
  - Q is the quantizer precision in bits,

I/O Bottleneck

- Decoder input bus width

$$W = \frac{\gamma \times Q}{R \times f_c}$$

where

  - γ is the throughput in b/s,
  - Q is the quantizer precision in bits,
  - R is the code rate, and

I/O Bottleneck

- Decoder input bus width

$$W = \frac{\gamma \times Q}{R \times f_c}$$

where

- γ is the throughput in b/s,
- Q is the quantizer precision in bits,
- R is the code rate, and
- f_c is the clock frequency.

I/O Bottleneck

- Decoder input bus width

$$W = \frac{\gamma \times Q}{R \times f_c}$$

  where

  - γ is the throughput in b/s,
  - Q is the quantizer precision in bits,
  - R is the code rate, and
  - f_c is the clock frequency.

- For $\gamma = 1$ Tb/s, $Q = 5$ bits, $R = 15/16$, and $f_c = 1$ GHz, the bus width is $W = 5333$ bits.

I/O Bottleneck

- Decoder input bus width

$$W = \frac{\gamma \times Q}{R \times f_c}$$

  where
  - γ is the throughput in b/s,
  - Q is the quantizer precision in bits,
  - R is the code rate, and
  - f_c is the clock frequency.
- For $\gamma = 1$ Tb/s, $Q = 5$ bits, $R = 15/16$, and $f_c = 1$ GHz, the bus width is $W = 5333$ bits.
- Better to use small Q and large R

I/O Bottleneck

- Decoder input bus width

$$W = \frac{\gamma \times Q}{R \times f_c}$$

  where
  - γ is the throughput in b/s,
  - Q is the quantizer precision in bits,
  - R is the code rate, and
  - f_c is the clock frequency.
- For $\gamma = 1$ Tb/s, $Q = 5$ bits, $R = 15/16$, and $f_c = 1$ GHz, the bus width is $W = 5333$ bits.
- Better to use small Q and large R
- Compress the LLR input as much as possible (syndrome techniques)

Memory bottleneck

- Decoder storage requirement

$$M = N \times P \times \overline{Q}$$

where

Memory bottleneck

- Decoder storage requirement

$$M = N \times P \times \overline{Q}$$

where

- N is the code block length,

Memory bottleneck

- Decoder storage requirement

$$M = N \times P \times \overline{Q}$$

where

  - N is the code block length,
  - P is the number of codewords being decoded in parallel

Memory bottleneck

- Decoder storage requirement

$$M = N \times P \times \overline{Q}$$

where

  - N is the code block length,
  - P is the number of codewords being decoded in parallel
  - $\overline{Q}$ is the average LLR precision

Memory bottleneck

- Decoder storage requirement

$$M = N \times P \times \overline{Q}$$

  where
  - N is the code block length,
  - P is the number of codewords being decoded in parallel
  - $\overline{Q}$ is the average LLR precision
- Use advanced quantization methods to reduce $\overline{Q}$

Memory bottleneck

- Decoder storage requirement

$$M = N \times P \times \overline{Q}$$

  where

  - N is the code block length,
  - P is the number of codewords being decoded in parallel
  - $\overline{Q}$ is the average LLR precision

- Use advanced quantization methods to reduce $\overline{Q}$
- Keep NP small

Polar code decoding algorithms

Algorithm Type	Computational Complexity	Space Complexity	Time Complexity
SC	$N \log N$	N	N
BP	$IN \log N$	$N \log N$	$I \log N$
SC-list	$LN \log N$	LN	$N + K \log^2(L)$
SC-stack	$DN \log N$	DN	-
SC-soft-out	$IN \log N$	$N \log N$	IN
SC-flip	$N \log N(1 + P_e(SNR))$	N	IN
MJL	$KN^{\log 3}$	N	$\log^2(N)$
Sphere	Cubic	-	-
I: number of iterations, L: list size, D: stack depth			

- Successive Cancellation (SC) decoder has the least computational and space complexity

Polar code decoding algorithms

Algorithm Type	Computational Complexity	Space Complexity	Time Complexity
SC	$N \log N$	N	N
BP	$IN \log N$	$N \log N$	$I \log N$
SC-list	$LN \log N$	LN	$N + K \log^2(L)$
SC-stack	$DN \log N$	DN	-
SC-soft-out	$IN \log N$	$N \log N$	IN
SC-flip	$N \log N(1 + P_e(SNR))$	N	IN
MJL	$KN^{\log 3}$	N	$\log^2(N)$
Sphere	Cubic	-	-
I: number of iterations, L: list size, D: stack depth			

- Successive Cancellation (SC) decoder has the least computational and space complexity
- Majority Logic (MJL) decoder has the least time complexity

Polar code decoding algorithms

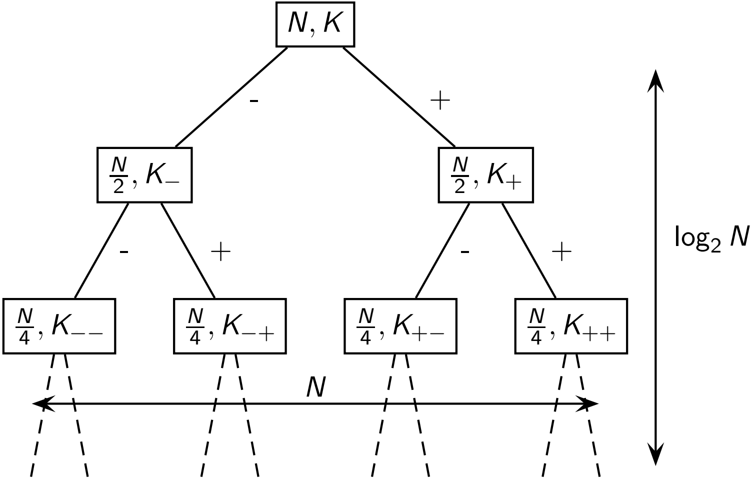
Algorithm Type	Computational Complexity	Space Complexity	Time Complexity
SC	$N \log N$	N	N
BP	$IN \log N$	$N \log N$	$I \log N$
SC-list	$LN \log N$	LN	$N + K \log^2(L)$
SC-stack	$DN \log N$	DN	-
SC-soft-out	$IN \log N$	$N \log N$	IN
SC-flip	$N \log N(1 + P_e(SNR))$	N	IN
MJL	$KN^{\log 3}$	N	$\log^2(N)$
Sphere	Cubic	-	-
I: number of iterations, L: list size, D: stack depth			

- Successive Cancellation (SC) decoder has the least computational and space complexity
- Majority Logic (MJL) decoder has the least time complexity
- We present a solution that combines the two approaches

SC Decoding Algorithm

$\mathcal{O}(N \log N)$ computational complexity

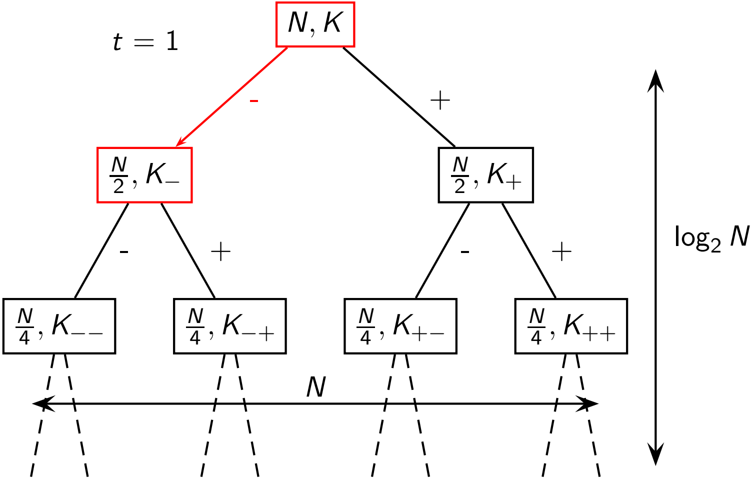
$\mathcal{O}(N)$ space and time complexity



SC Decoding Algorithm

$\mathcal{O}(N \log N)$ computational complexity

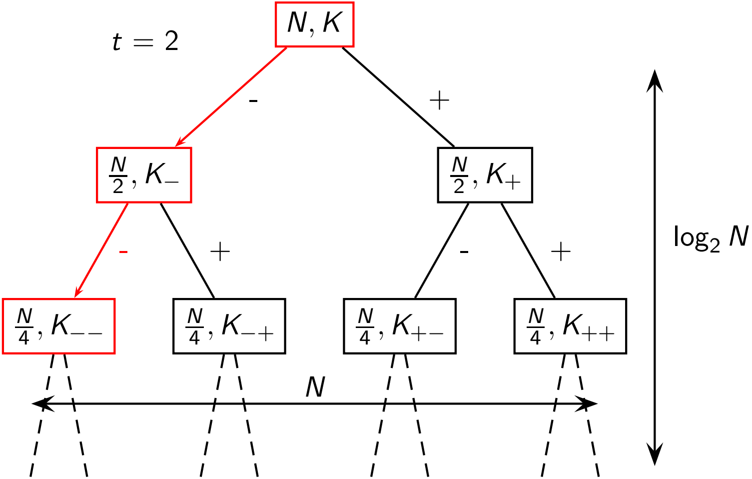
$\mathcal{O}(N)$ space and time complexity



SC Decoding Algorithm

$\mathcal{O}(N \log N)$ computational complexity

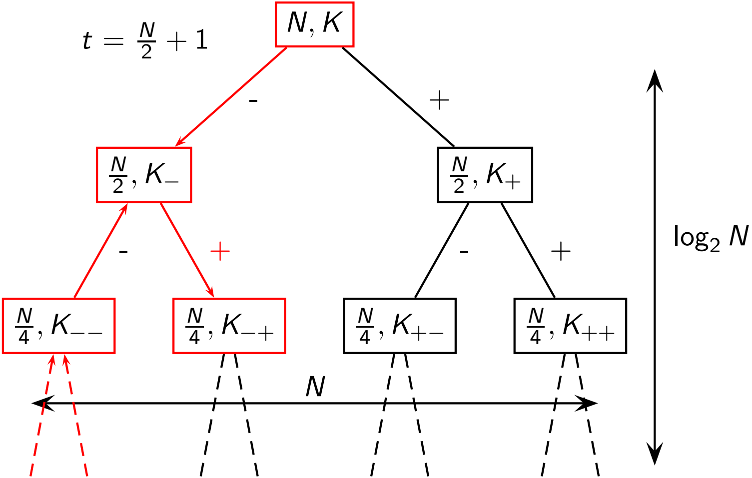
$\mathcal{O}(N)$ space and time complexity



SC Decoding Algorithm

$\mathcal{O}(N \log N)$ computational complexity

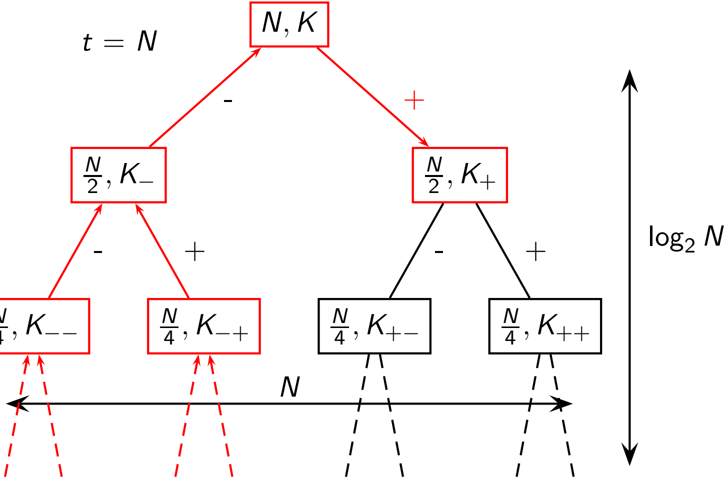
$\mathcal{O}(N)$ space and time complexity



SC Decoding Algorithm

$\mathcal{O}(N \log N)$ computational complexity

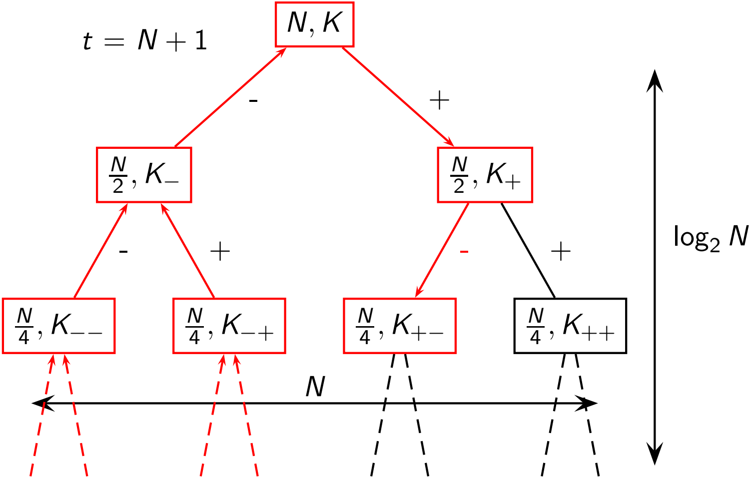
$\mathcal{O}(N)$ space and time complexity



SC Decoding Algorithm

$\mathcal{O}(N \log N)$ computational complexity

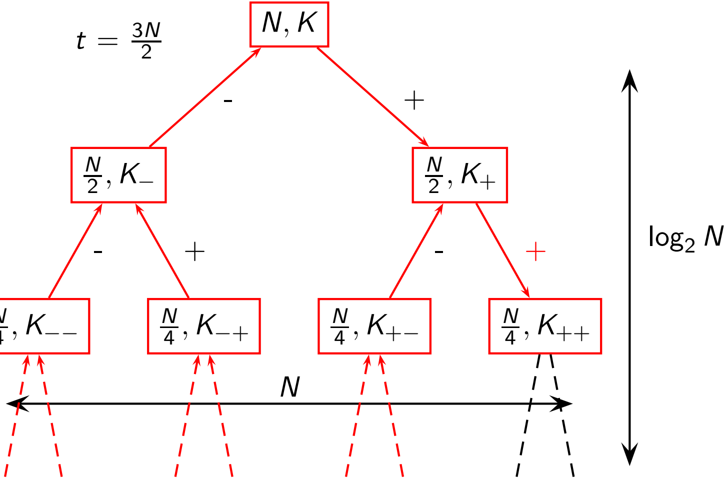
$\mathcal{O}(N)$ space and time complexity



SC Decoding Algorithm

$\mathcal{O}(N \log N)$ computational complexity

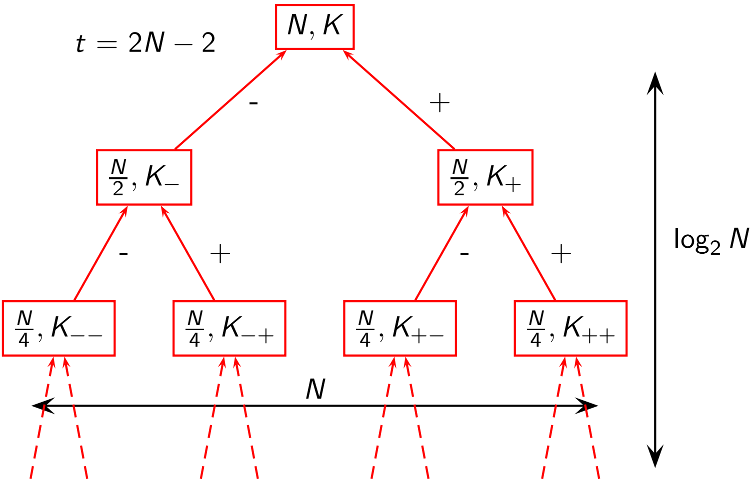
$\mathcal{O}(N)$ space and time complexity



SC Decoding Algorithm

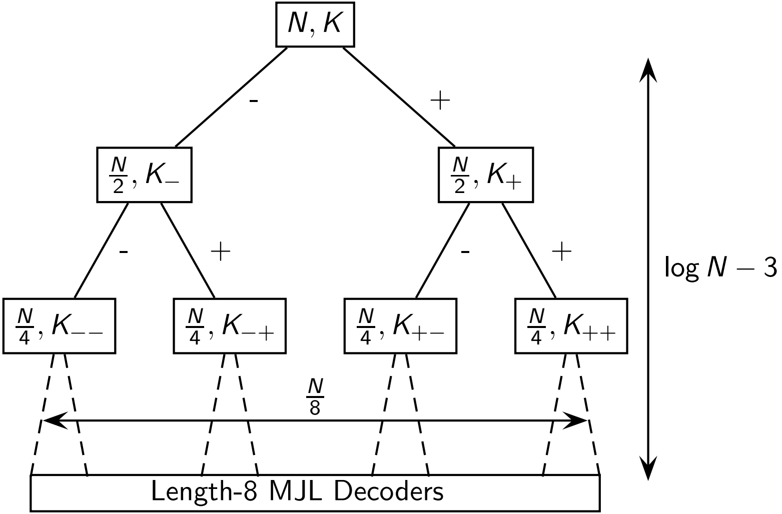
$\mathcal{O}(N \log N)$ computational complexity

$\mathcal{O}(N)$ space and time complexity



SC-MJL Decoding Algorithm

- Use length-8 MJL decoders for $N \leq 8$ polar codes to reduce the latency of pure SC decoding from $2N - 2$ to $\frac{9N}{8} - 2$



SC-MJL Decoding Algorithm (Continued)

- Use SC decoder to decompose a length-N polar code into two length-$N/2$ polar codes until length $N = 8$ is reached.

SC-MJL Decoding Algorithm (Continued)

- Use SC decoder to decompose a length-N polar code into two length-$N/2$ polar codes until length $N = 8$ is reached.
- At length $N = 8$

SC-MJL Decoding Algorithm (Continued)

- Use SC decoder to decompose a length-N polar code into two length-$N/2$ polar codes until length $N = 8$ is reached.
- At length $N = 8$
  - use the Wagner decoder for single-parity-check codes

SC-MJL Decoding Algorithm (Continued)

- Use SC decoder to decompose a length-N polar code into two length-$N/2$ polar codes until length $N = 8$ is reached.
- At length $N = 8$
  - use the Wagner decoder for single-parity-check codes
  - decode rate-0 and rate-1 codes trivially

SC-MJL Decoding Algorithm (Continued)

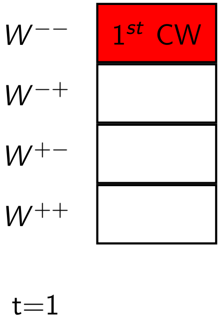
- Use SC decoder to decompose a length-N polar code into two length-$N/2$ polar codes until length $N = 8$ is reached.
- At length $N = 8$
  - use the Wagner decoder for single-parity-check codes
  - decode rate-0 and rate-1 codes trivially
  - use ML decoder for repetition codes

SC-MJL Decoding Algorithm (Continued)

- Use SC decoder to decompose a length-N polar code into two length-$N/2$ polar codes until length $N = 8$ is reached.
- At length $N = 8$
  - use the Wagner decoder for single-parity-check codes
  - decode rate-0 and rate-1 codes trivially
  - use ML decoder for repetition codes
  - use MJL decoder for all remaining length-8 codes

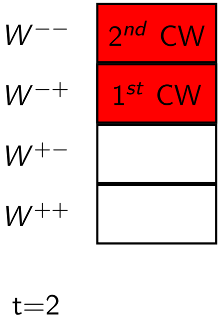
Pipelined Decoding

- Decoder can take a new CW at each time interval represented by clock cycles



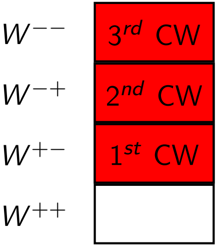
Pipelined Decoding

- Decoder can take a new CW at each time interval represented by clock cycles
- Pipelining increases both hardware efficiency and power density



Pipelined Decoding

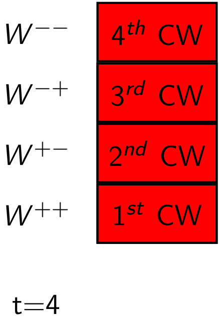
- Decoder can take a new CW at each time interval represented by clock cycles
- Pipelining increases both hardware efficiency and power density



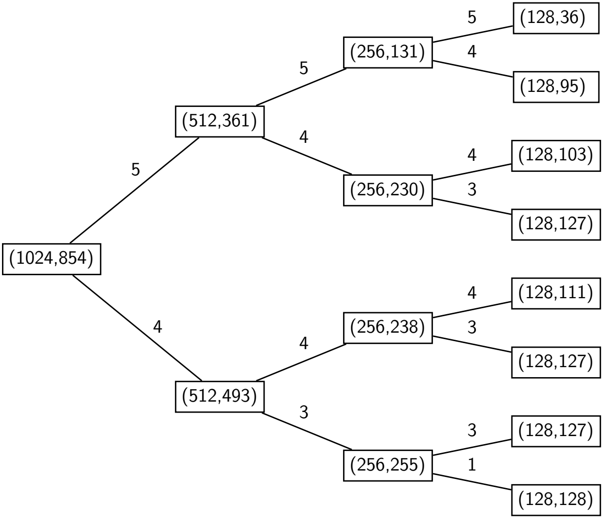
t=3

Pipelined Decoding

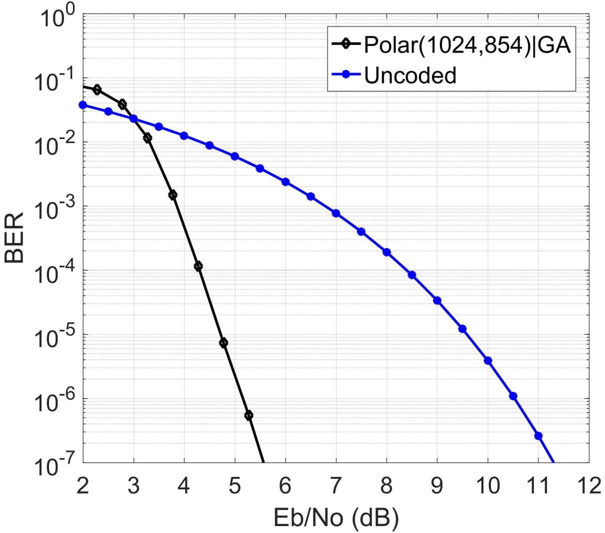
- Decoder can take a new CW at each time interval represented by clock cycles
- Pipelining increases both hardware efficiency and power density



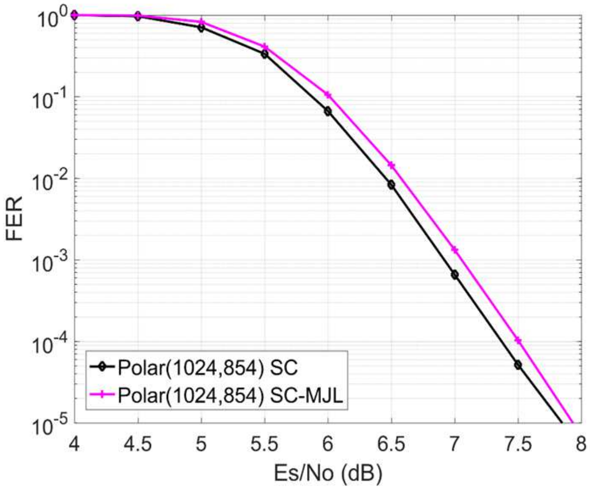
Progressive quantization of LLRs inside the decoder



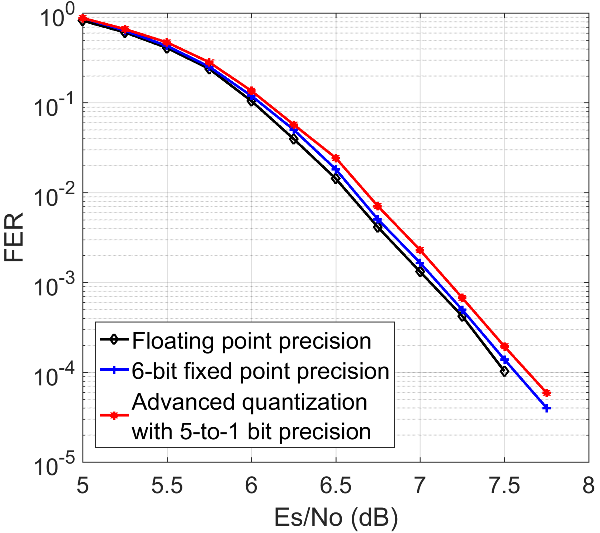
SC Performance



SC-MJL Performance



Effect of progressive quantization



Post-synthesis results at 7nm

Decoding Algorithm	SC	SC-MJL	SC-MJL	SC-MJL
Quantization (bits)	6	6	5-to-1	5-to-1
Combining used?	×	×	×	✓
Throughput (Gb/s)	1000			
Area (mm^2)	10			
Area Eff. ($Gb/s/mm^2$)	100			
Pow. Den. (W/mm^2)	0.19	0.13	0.10	0.04
Energy Eff. (pJ/bit)	1.90	1.28	0.96	0.42
Latency (Clock cycles)	157	127	127	40
Freq. (MHz)	585.5			

- Two independent pipelined decoders are used each operating at 500 Gb/s

Post-synthesis results at 7nm

Decoding Algorithm	SC	SC-MJL	SC-MJL	SC-MJL
Quantization (bits)	6	6	5-to-1	5-to-1
Combining used?	×	×	×	✓
Throughput (Gb/s)	1000			
Area (mm^2)	10			
Area Eff. ($Gb/s/mm^2$)	100			
Pow. Den. (W/mm^2)	0.19	0.13	0.10	0.04
Energy Eff. (pJ/bit)	1.90	1.28	0.96	0.42
Latency (Clock cycles)	157	127	127	40
Freq. (MHz)	585.5			

- Two independent pipelined decoders are used each operating at 500 Gb/s
- Memory dominates the area and energy efficiency

Summary

- 1 Tb/s FEC appears feasible with 7nm technology with a 6 dB coding gain

Summary

- 1 Tb/s FEC appears feasible with 7nm technology with a 6 dB coding gain
- The proposed solution brought together some existing techniques

Summary

- 1 Tb/s FEC appears feasible with 7nm technology with a 6 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 7nm technology with a 6 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 7nm technology with a 6 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
  - Adaptive quantization to reduce memory usage

Summary

- 1 Tb/s FEC appears feasible with 7nm technology with a 6 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
  - Adaptive quantization to reduce memory usage
  - Reducing pipeline depth by combining simple steps

Summary

- 1 Tb/s FEC appears feasible with 7nm technology with a 6 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
  - Adaptive quantization to reduce memory usage
  - Reducing pipeline depth by combining simple steps
- Storage complexity dominates the design

Final remarks

- With VLSI technology reaching its limits, FEC designers will have to learn more about VLSI implementation constraints

Final remarks

- With VLSI technology reaching its limits, FEC designers will have to learn more about VLSI implementation constraints
- The situation is reminiscent of the first three decades of coding when hardware complexity was a binding constraint but hardware at that time was very simple

Final remarks

- With VLSI technology reaching its limits, FEC designers will have to learn more about VLSI implementation constraints
- The situation is reminiscent of the first three decades of coding when hardware complexity was a binding constraint but hardware at that time was very simple
- The discrepancy between the desired data rates and available clock frequency may never have been as high as today

Final remarks

- With VLSI technology reaching its limits, FEC designers will have to learn more about VLSI implementation constraints
- The situation is reminiscent of the first three decades of coding when hardware complexity was a binding constraint but hardware at that time was very simple
- The discrepancy between the desired data rates and available clock frequency may never have been as high as today
- We may hope that new types codes will emerge as we understand VLSI complexity vs FEC performance better

Acknowledgments

- This work has been carried out in part by support from EPIC project, with funding from the European Union's Horizon 2020 research and innovation programme under grant No. 760150.
- We thank Y. Ertuğrul for help with simulations and figures.

Thank you!