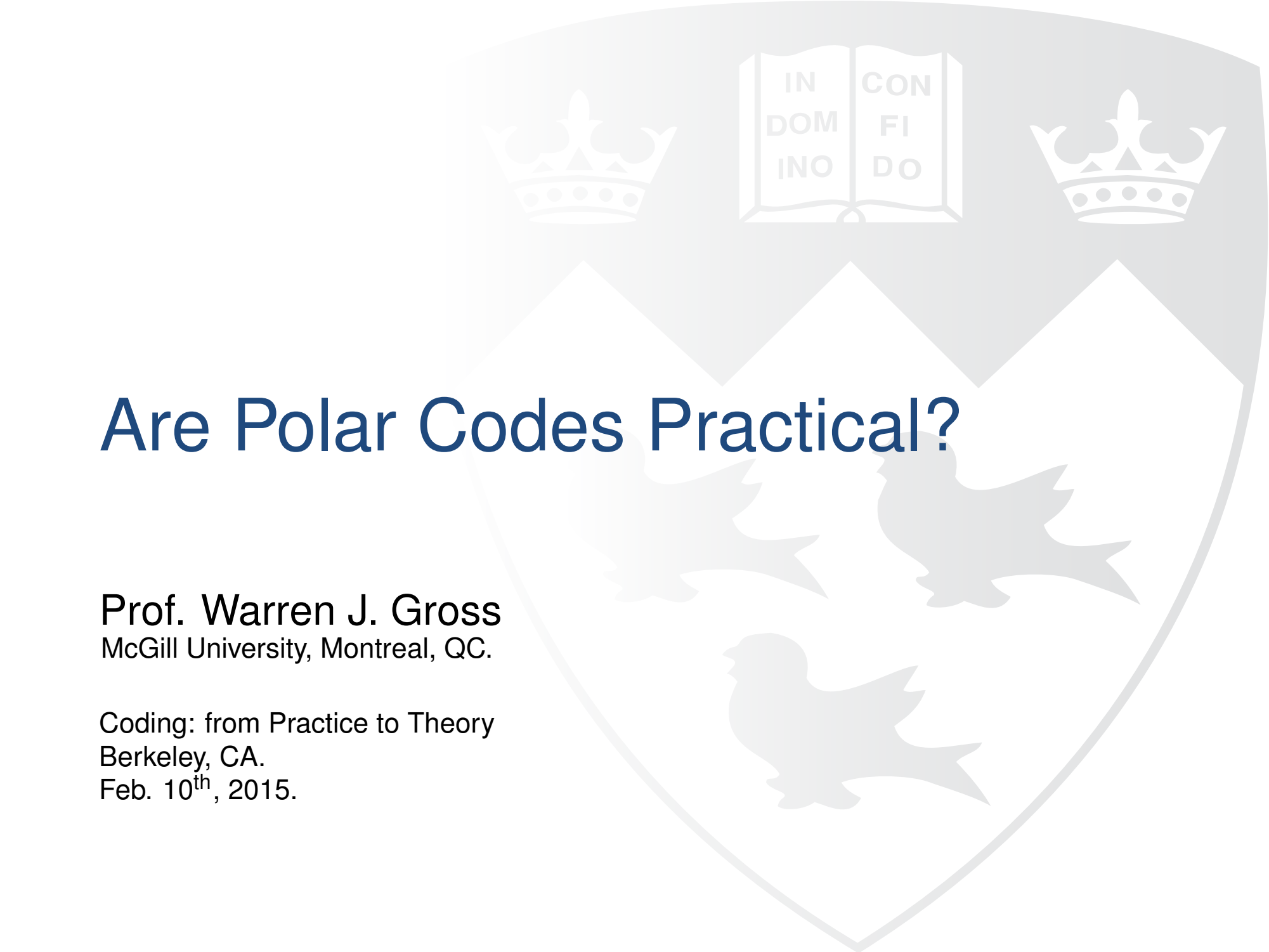


Are Polar Codes Practical?

Prof. Warren J. Gross
McGill University, Montreal, QC.

Coding: from Practice to Theory
Berkeley, CA.
Feb. 10th, 2015.

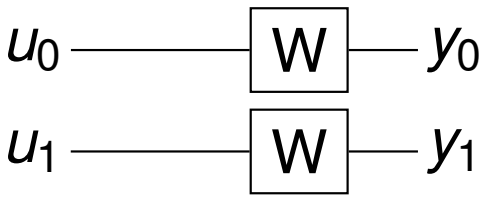
Achieving the Channel Capacity

Successive cancellation → low throughput.

Polar Codes
2008
0 dB
using SC
as $n \to \infty$

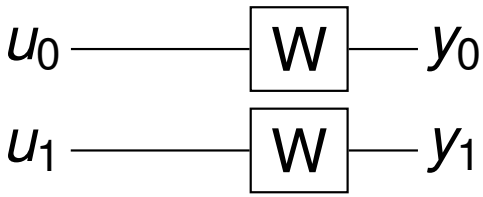
Mediocre performance at moderate code-length.
Becomes excellent for $N > 2^{20}$.

Channel Polarization

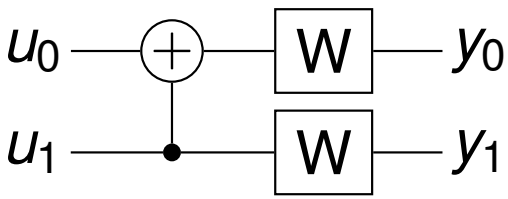


$$I^0 = I(W) = I^1$$

Channel Polarization

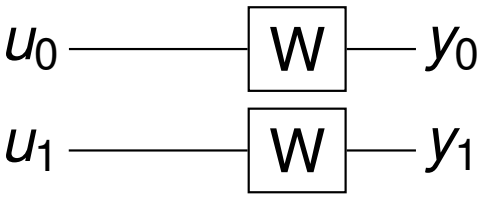


$$I^0 = I(W) = I^1$$

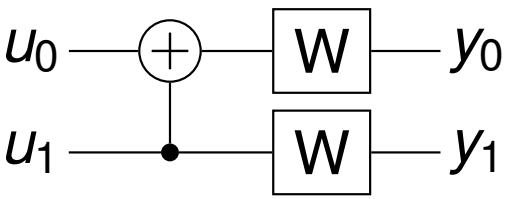


$$I^0 < I(W) < I^1$$

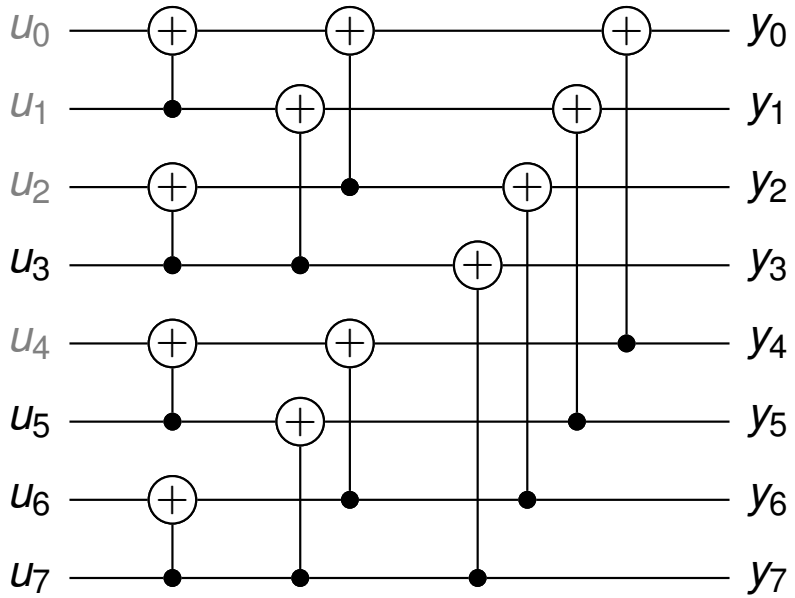
Channel Polarization



$$I^0 = I(W) = I^1$$

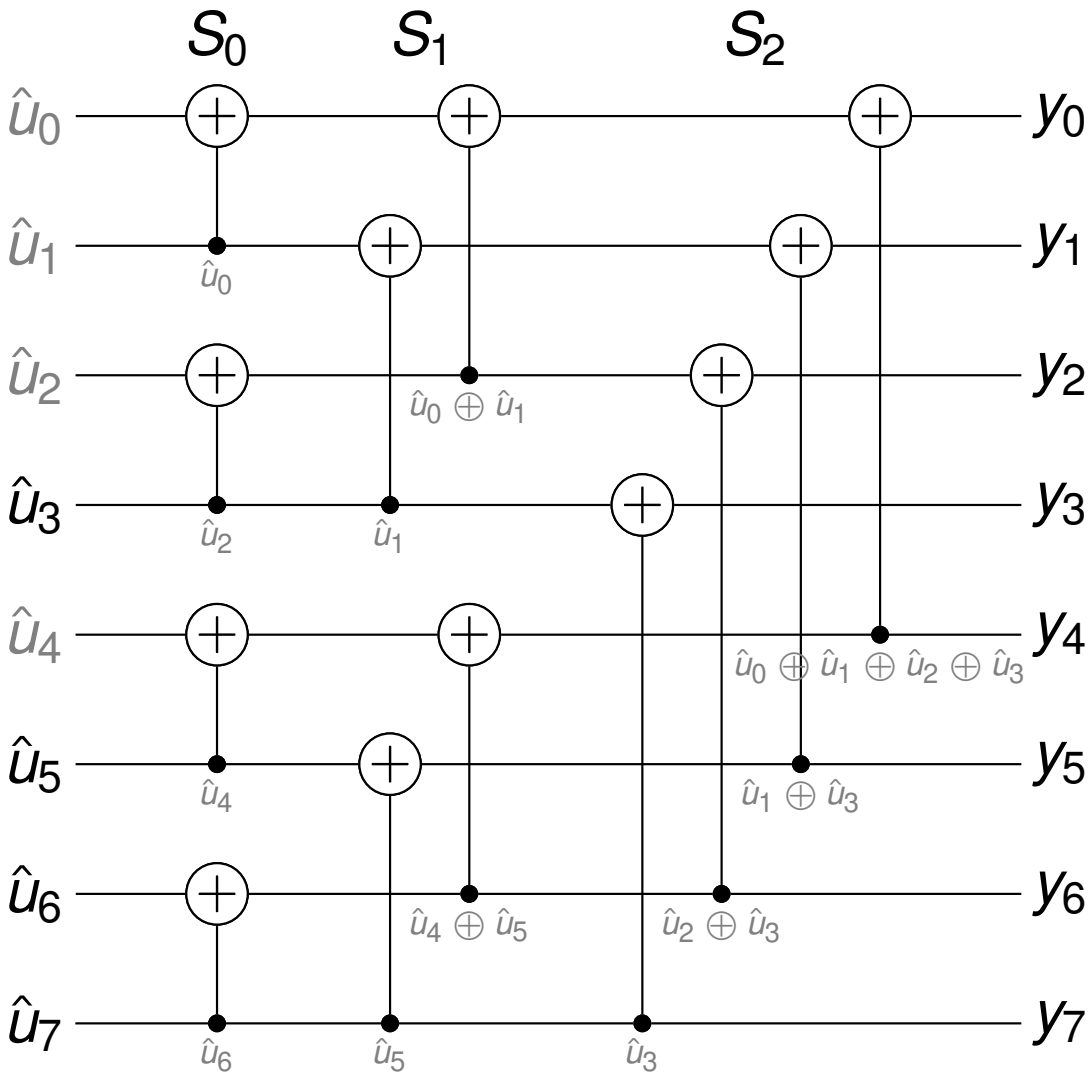


$$I^0 < I(W) < I^1$$

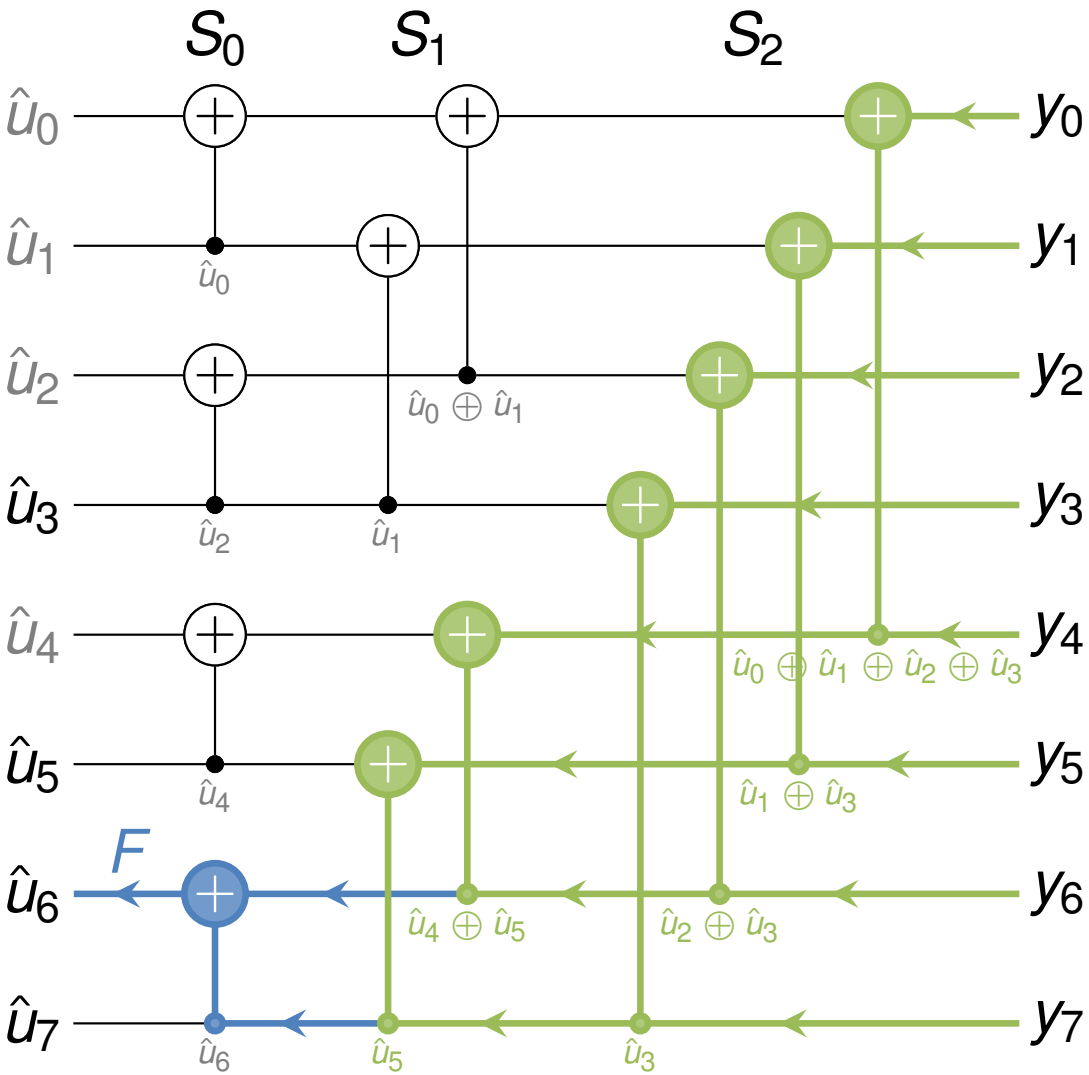


Proportion of reliable bits $\to I(W)$.

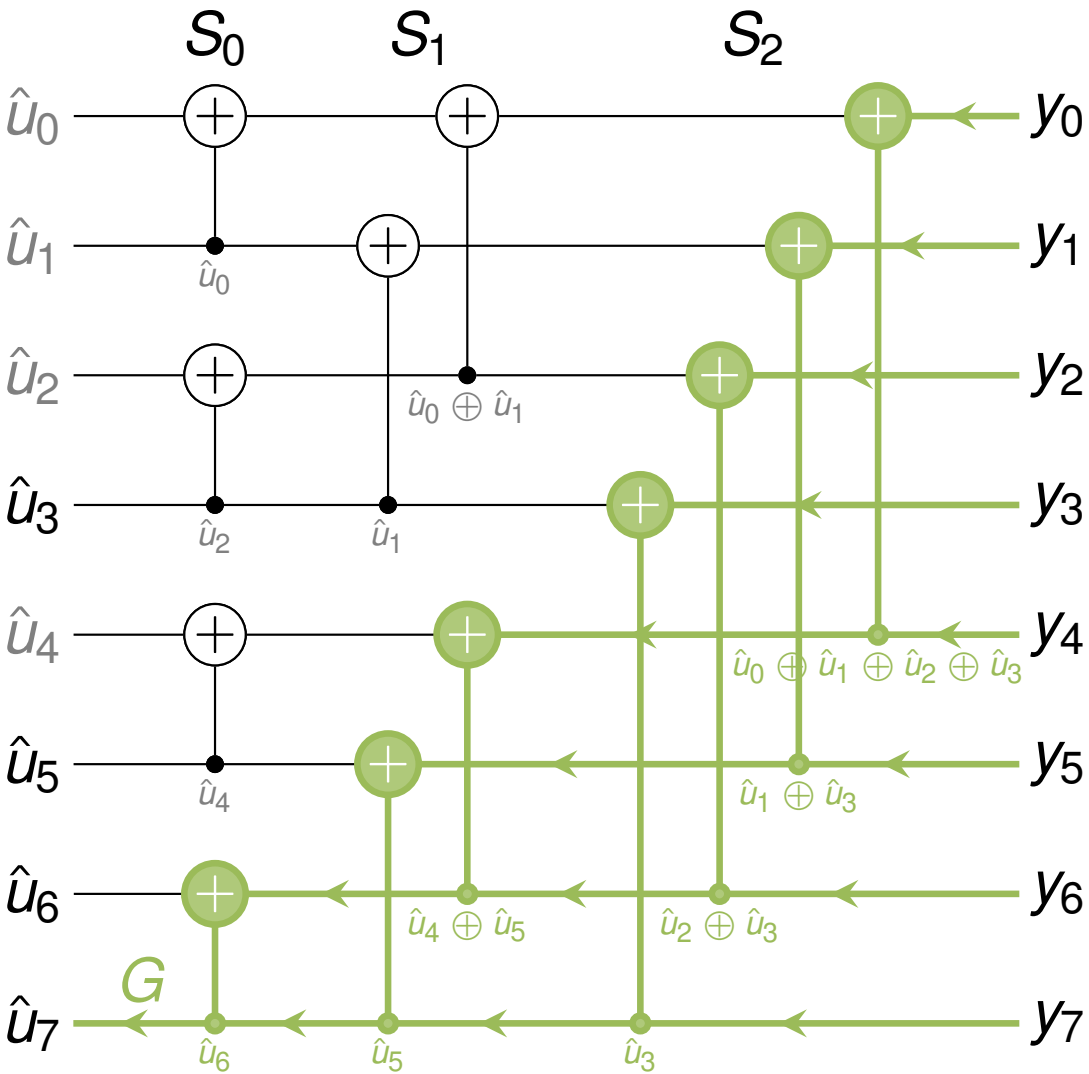
Successive Cancellation Decoding



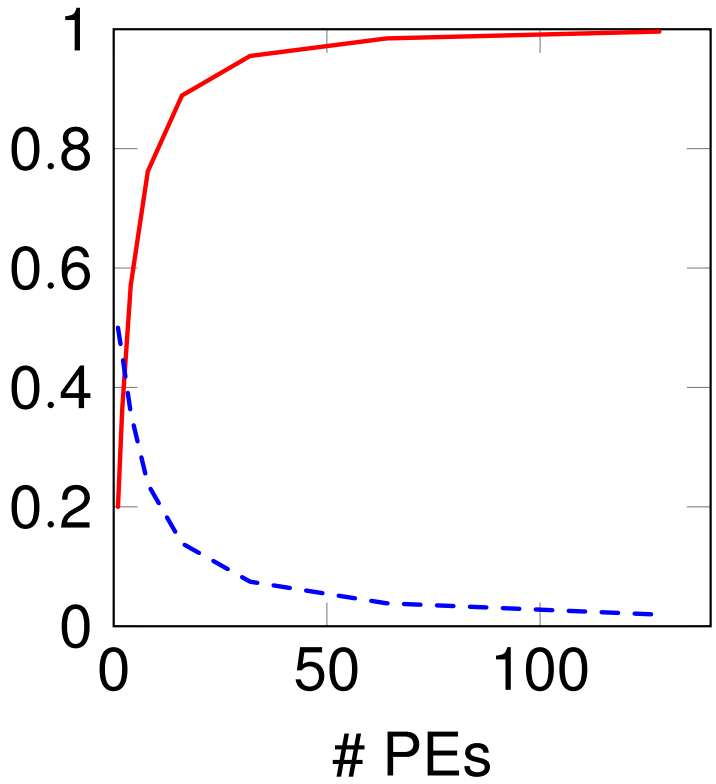
Successive Cancellation Decoding



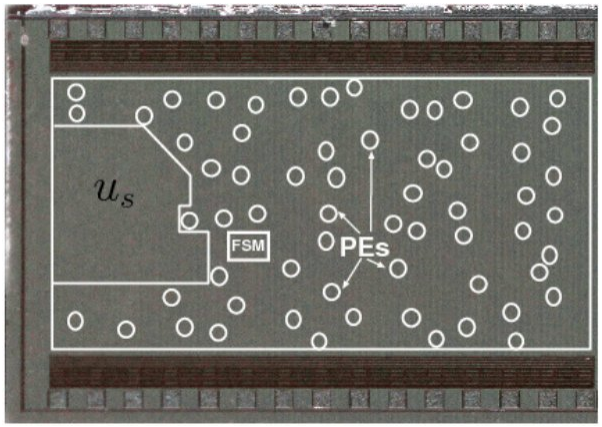
Successive Cancellation Decoding



Semi-Parallel SC Decoder Implementation



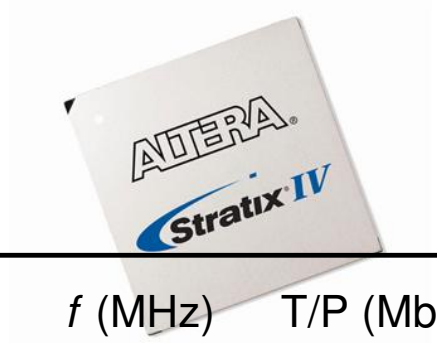
$>$ 95% speed with 64 PEs.



First polar decoder ASIC.

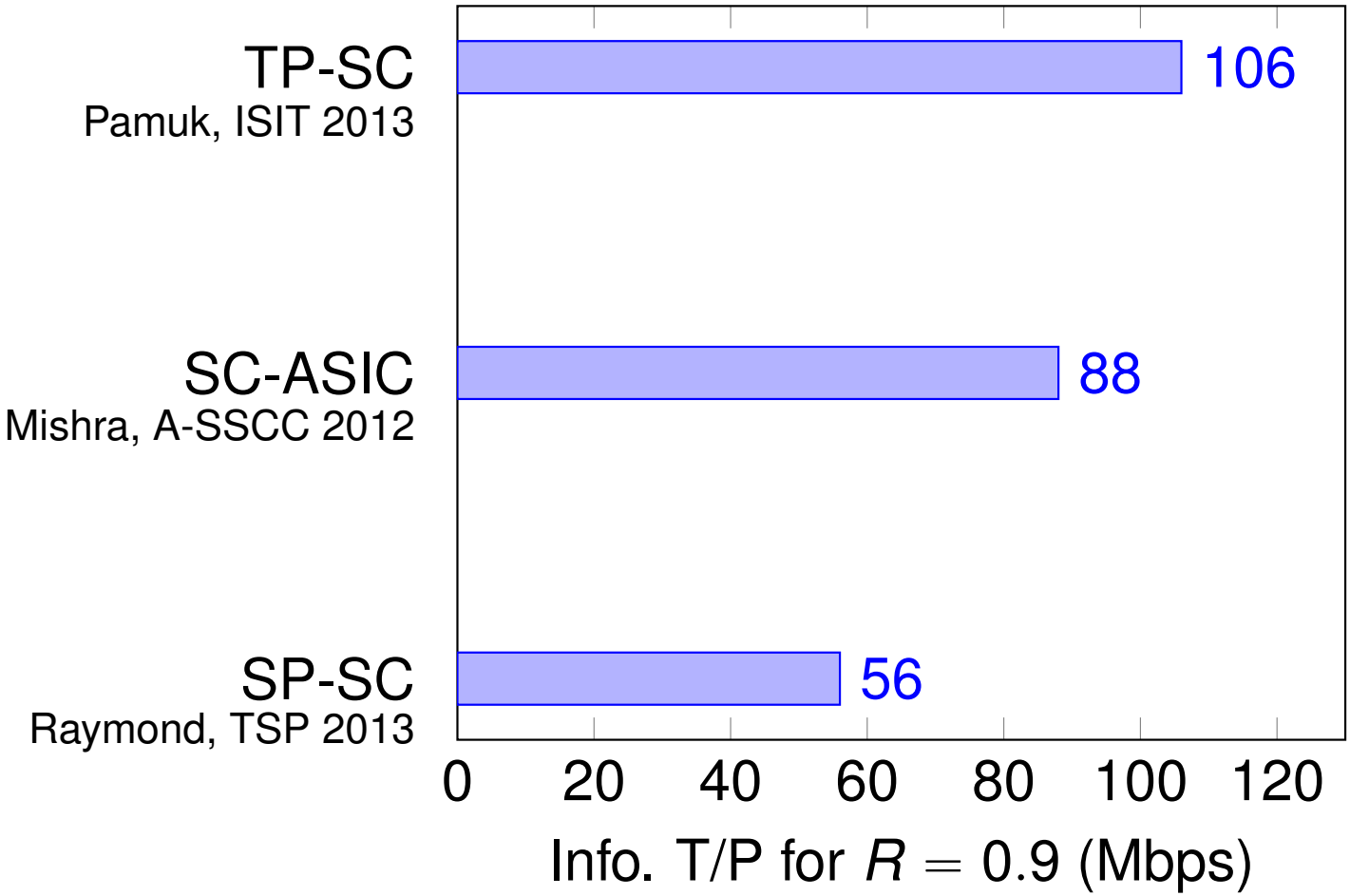
Code	(1024, 512)
Technology	180 nm
Area	1.72 mm^2
Frequency	150 MHz
Info. T/P	49 Mbps
Power	67 mW

FPGA Implementation



N	LUT	FF	RAM (bits)	f (MHz)	T/P (Mbps)
2^{15}	3,263	1,304	411,648	167	$62R$
2^{16}	3,414	1,316	821,248	157	$57R$
2^{18}	3,548	1,349	3,278,848	140	$51R$
2^{20}	5,956	1,366	13,109,248	102	$38R$

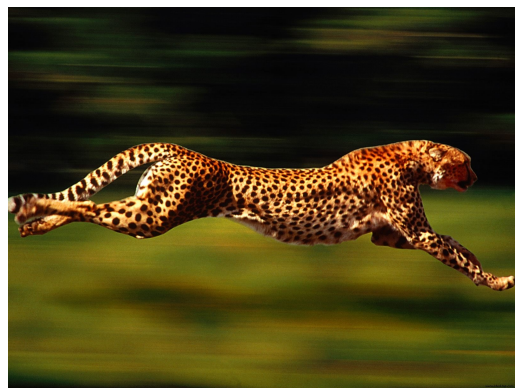
Throughput of SC Decoders



Improving Throughput

100 Mbps
SC

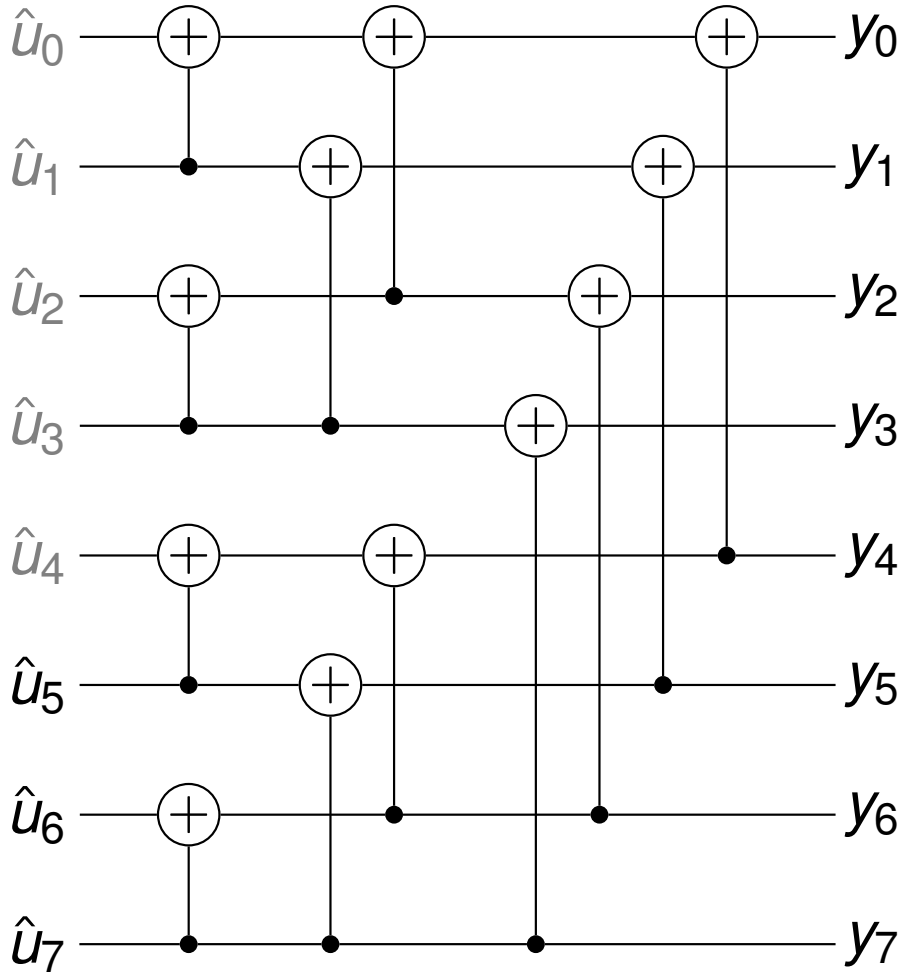
1000×



100 Gbps
Unrolled

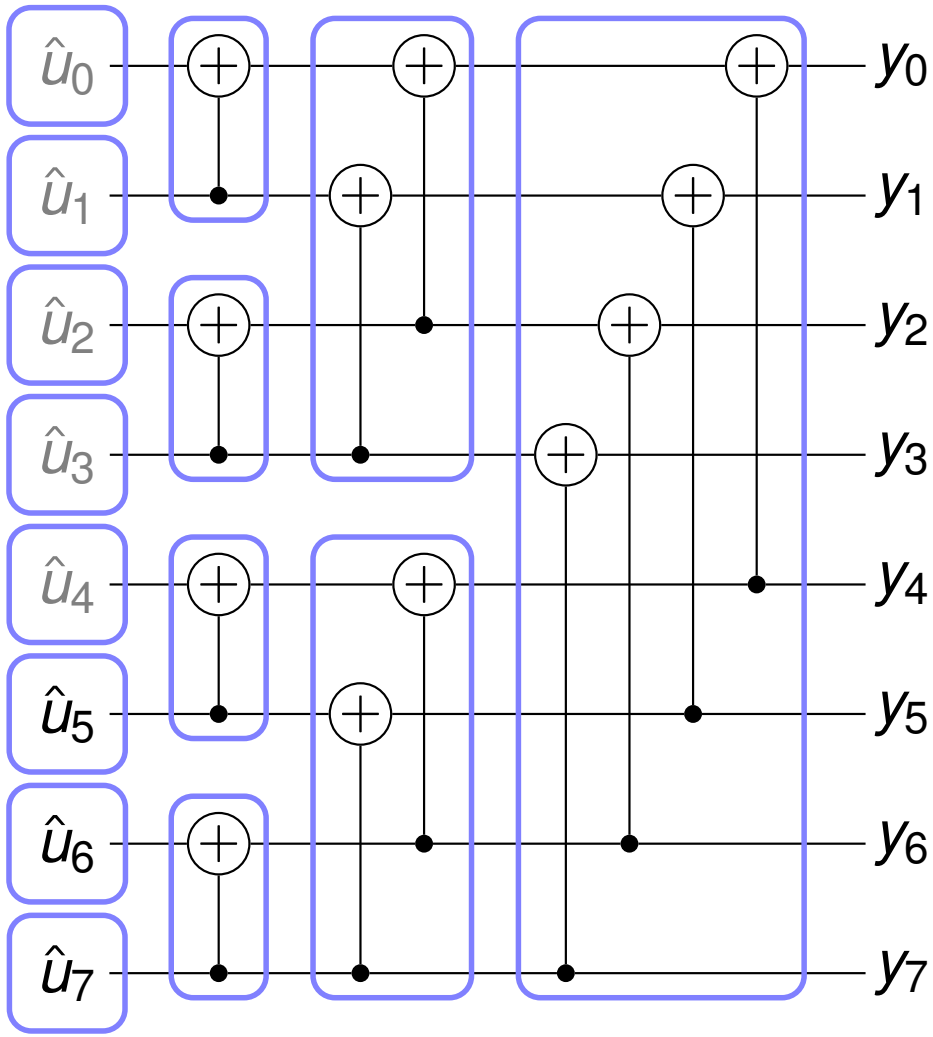
Successive Cancellation Tree

- View the SC decoder graph as a tree.



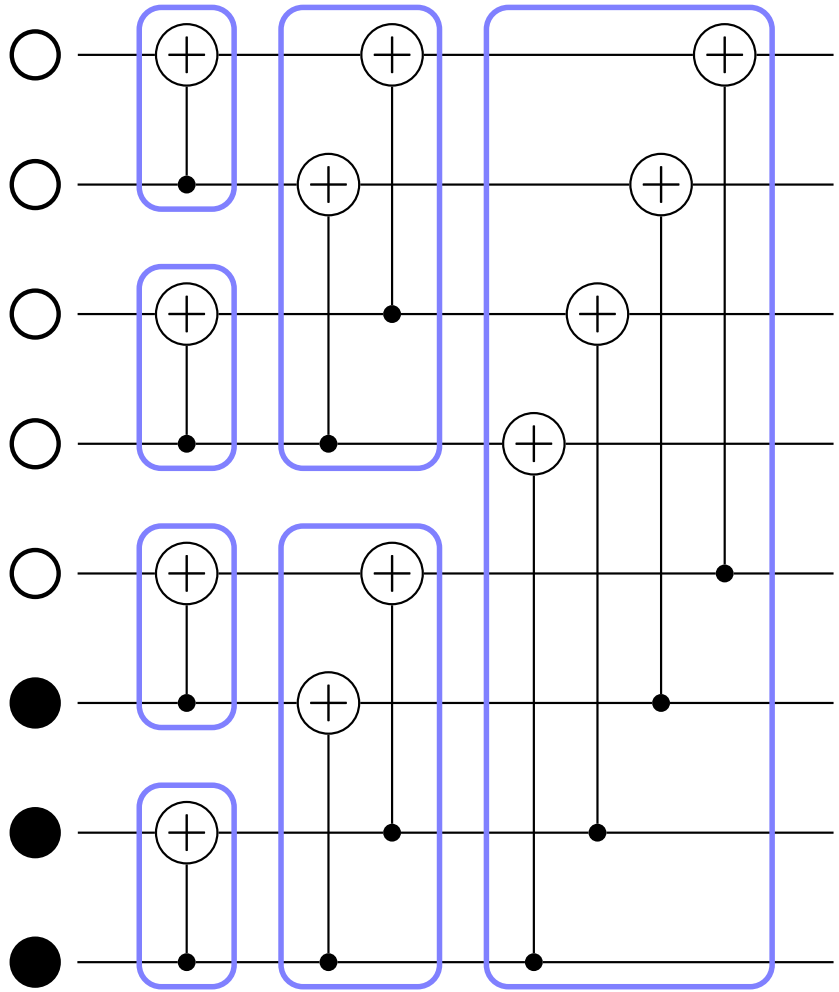
Successive Cancellation Tree

- View the SC decoder graph as a tree.



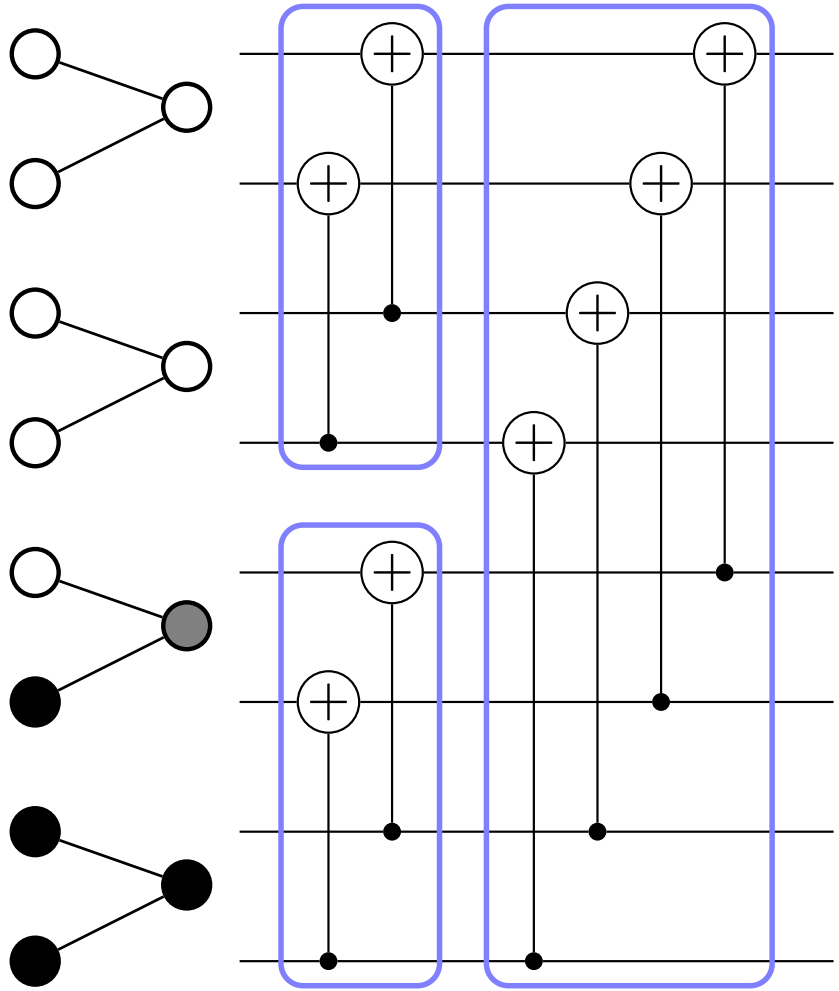
Successive Cancellation Tree

- View the SC decoder graph as a tree.



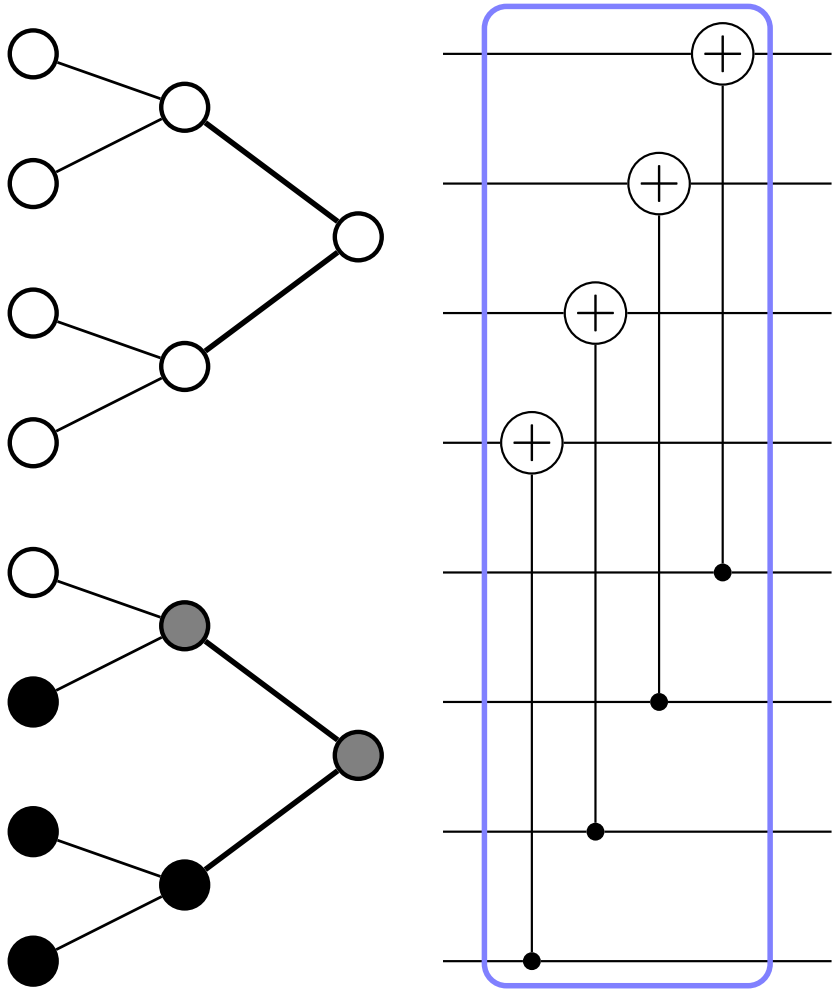
Successive Cancellation Tree

- View the SC decoder graph as a tree.



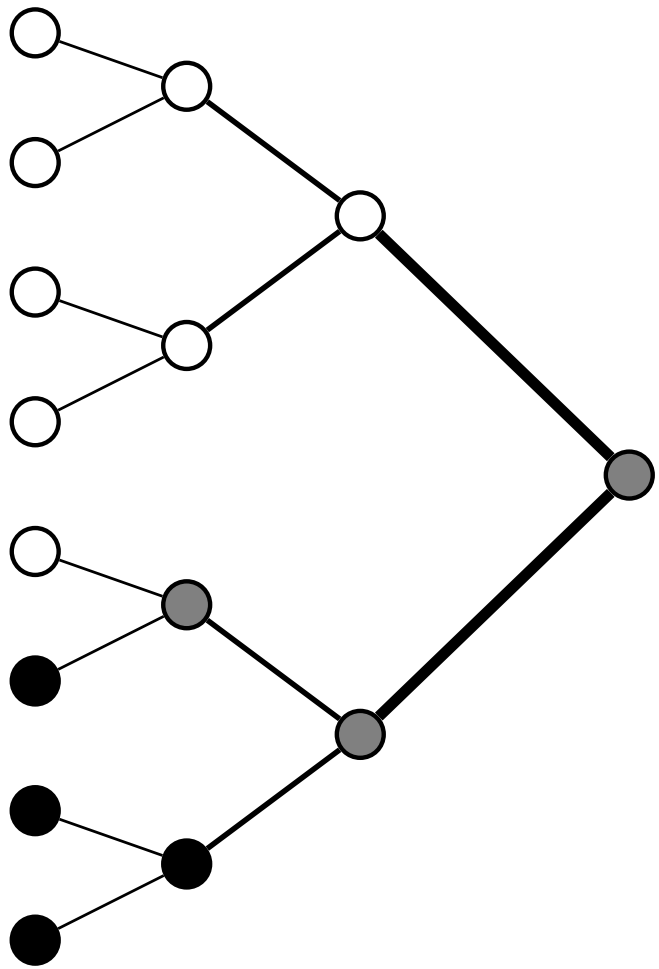
Successive Cancellation Tree

- View the SC decoder graph as a tree.

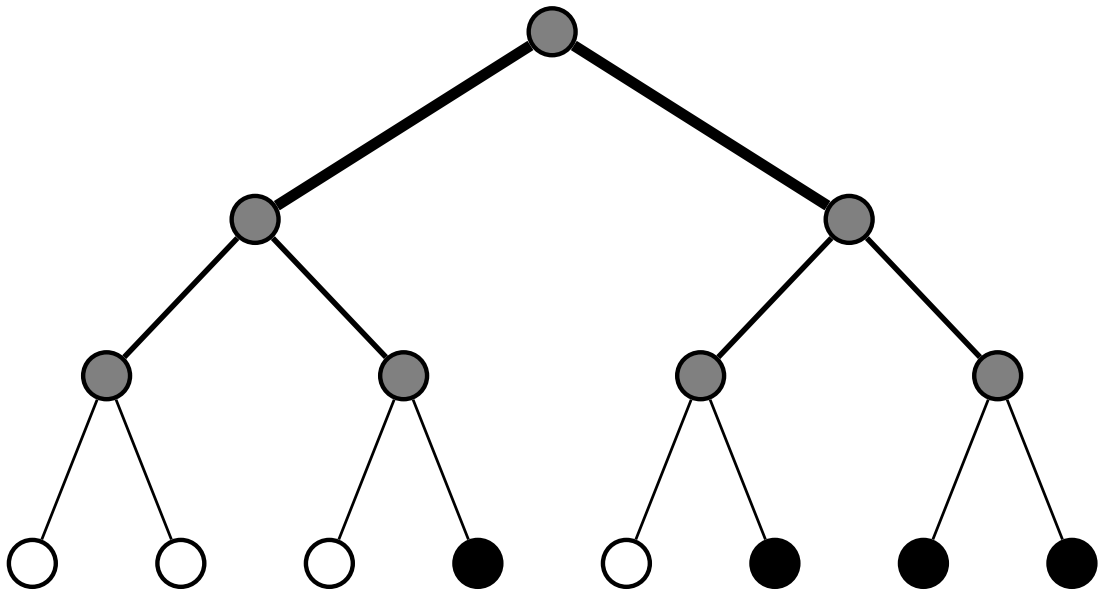


Successive Cancellation Tree

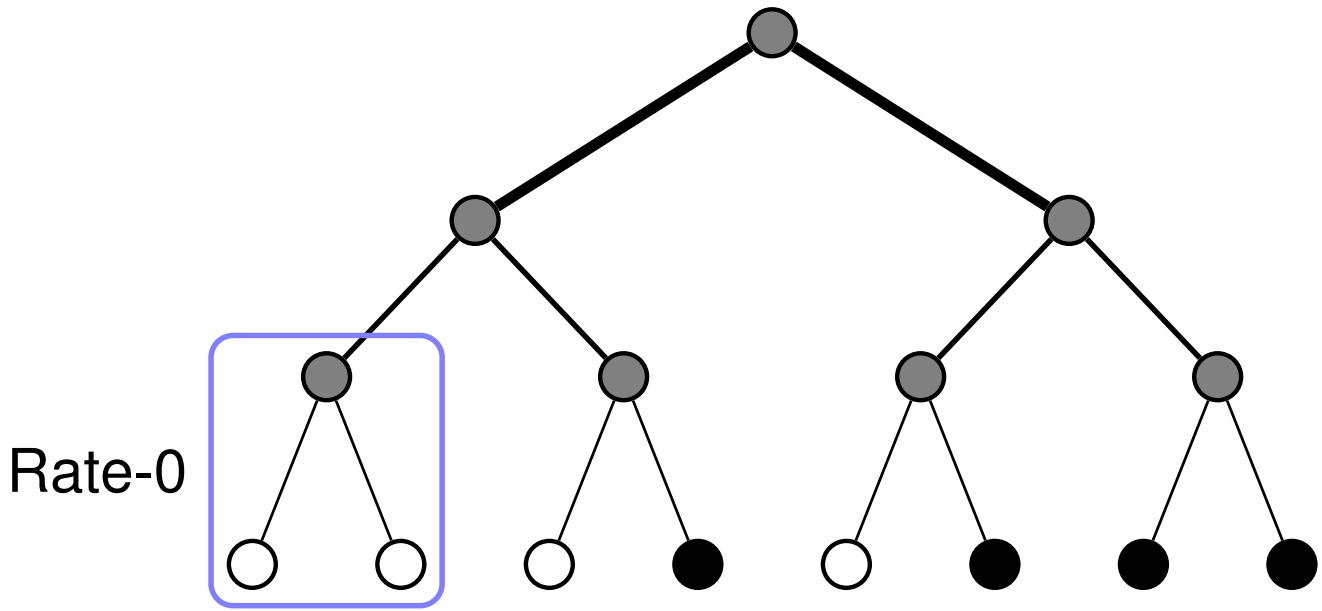
- View the SC decoder graph as a tree.



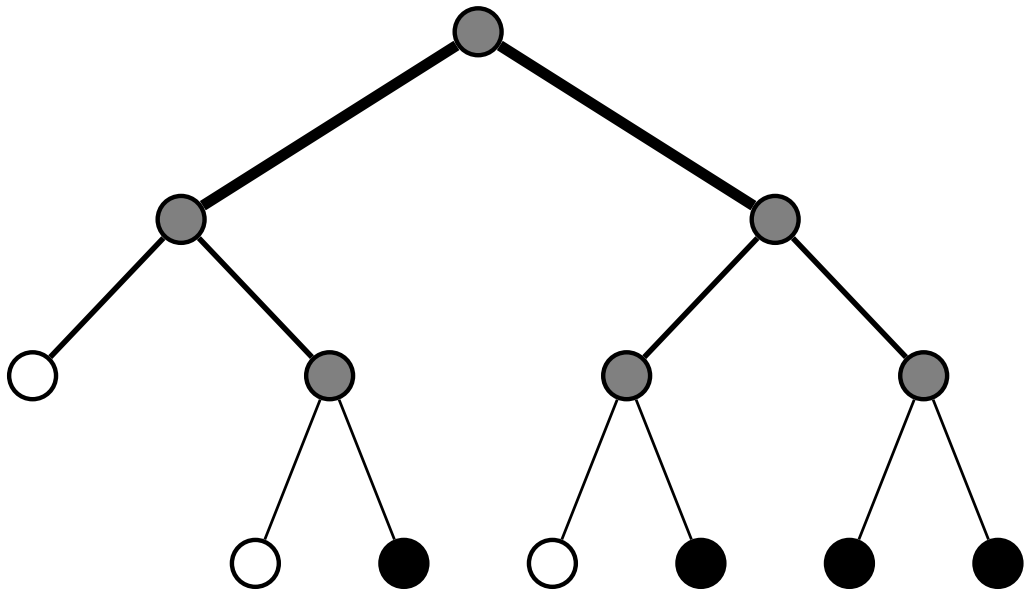
Simplified Successive-Cancellation Decoding



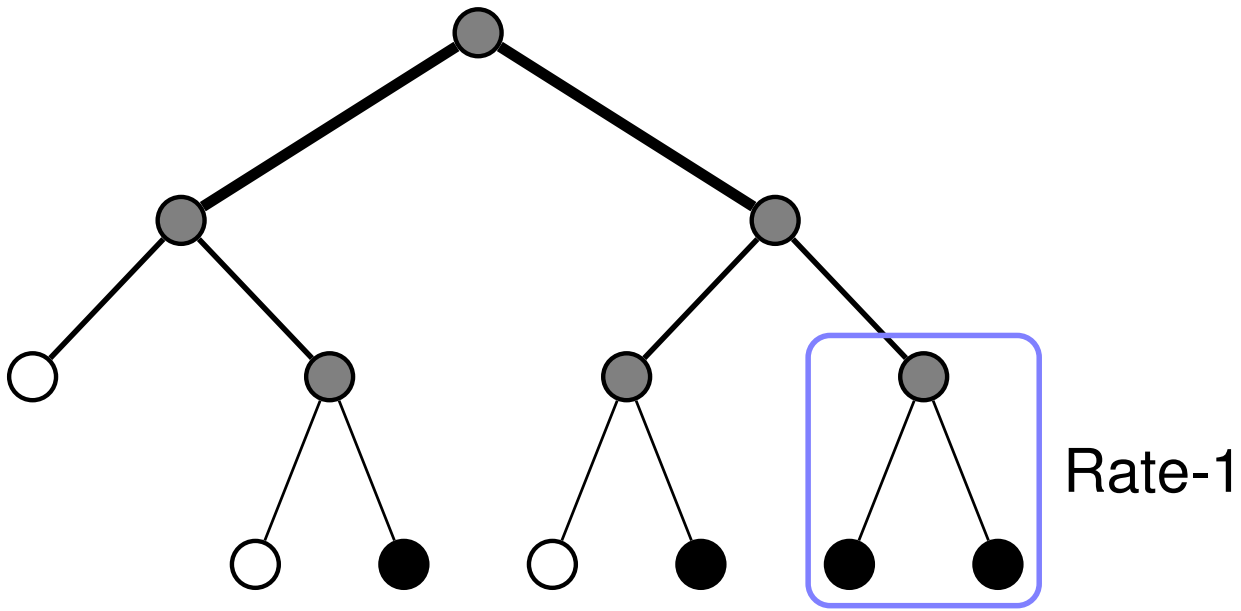
Simplified Successive-Cancellation Decoding



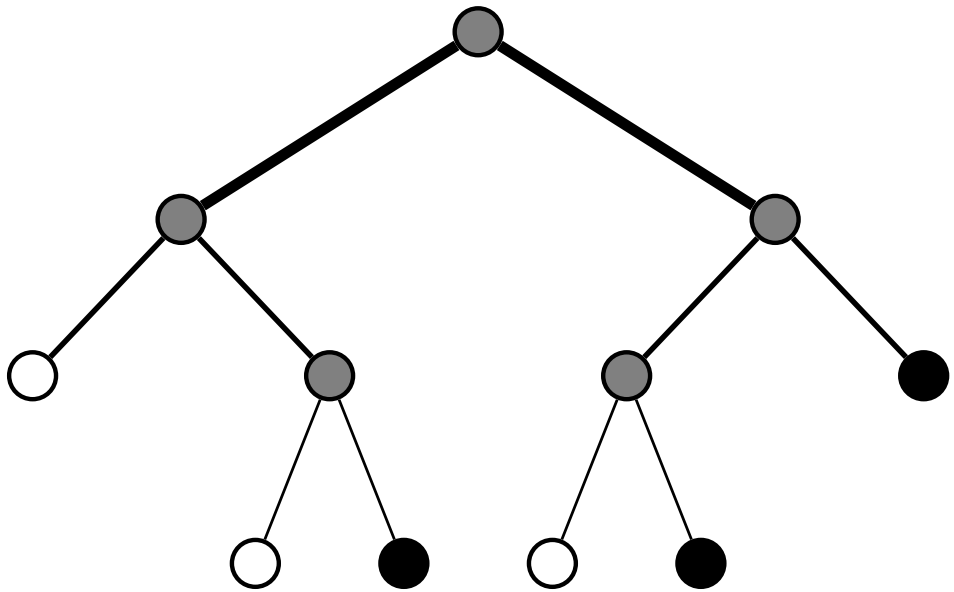
Simplified Successive-Cancellation Decoding



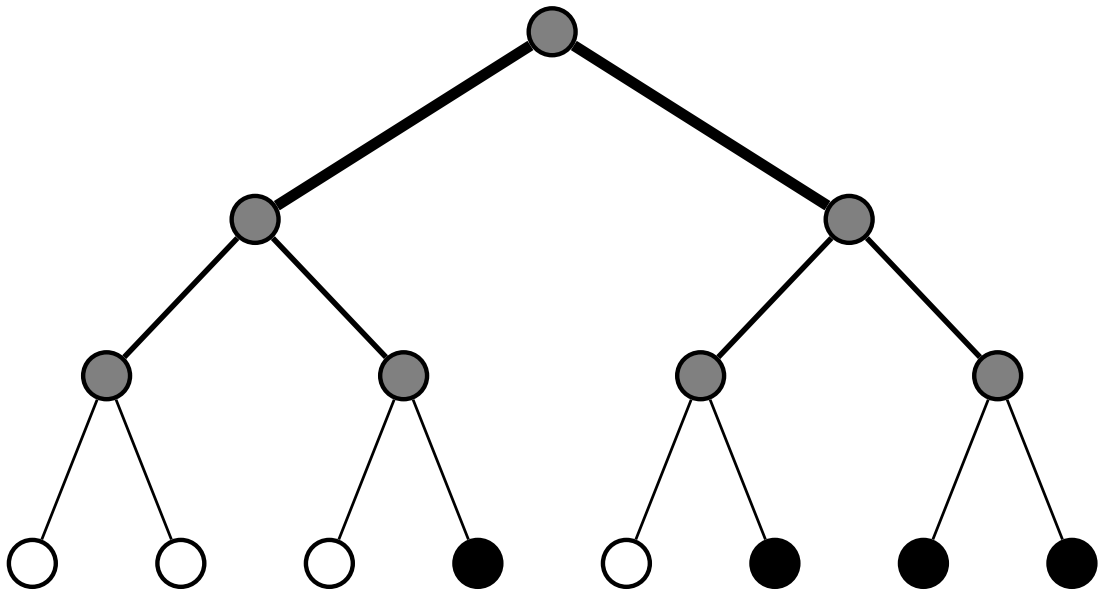
Simplified Successive-Cancellation Decoding



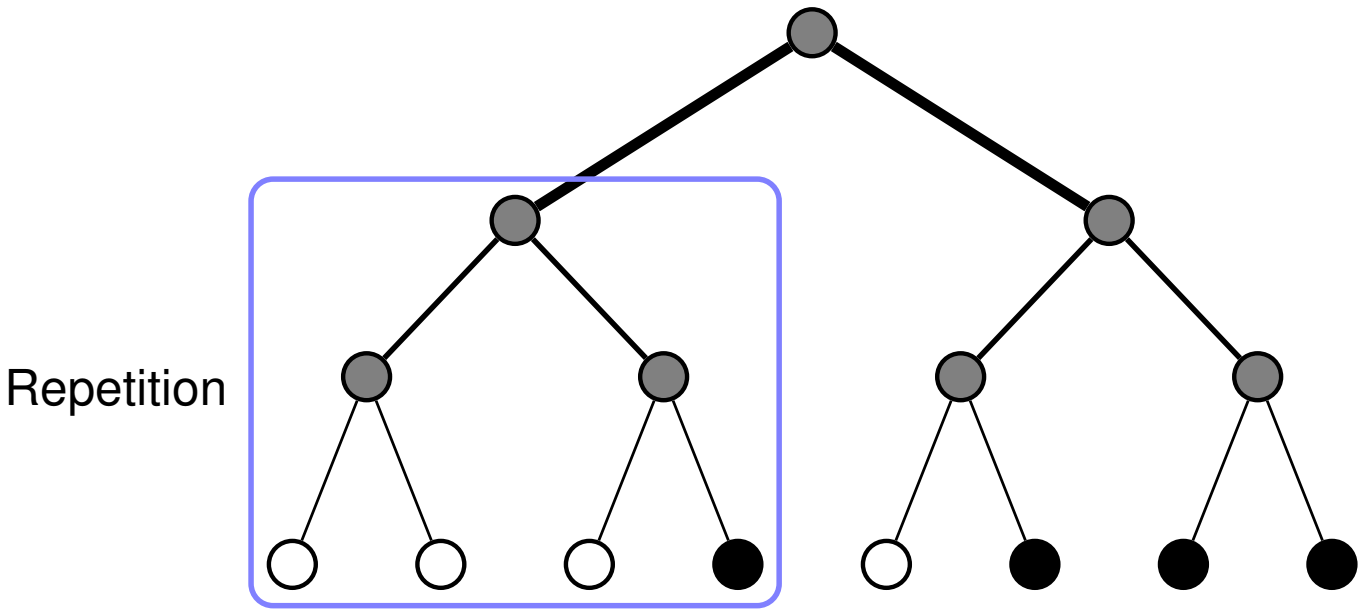
Simplified Successive-Cancellation Decoding



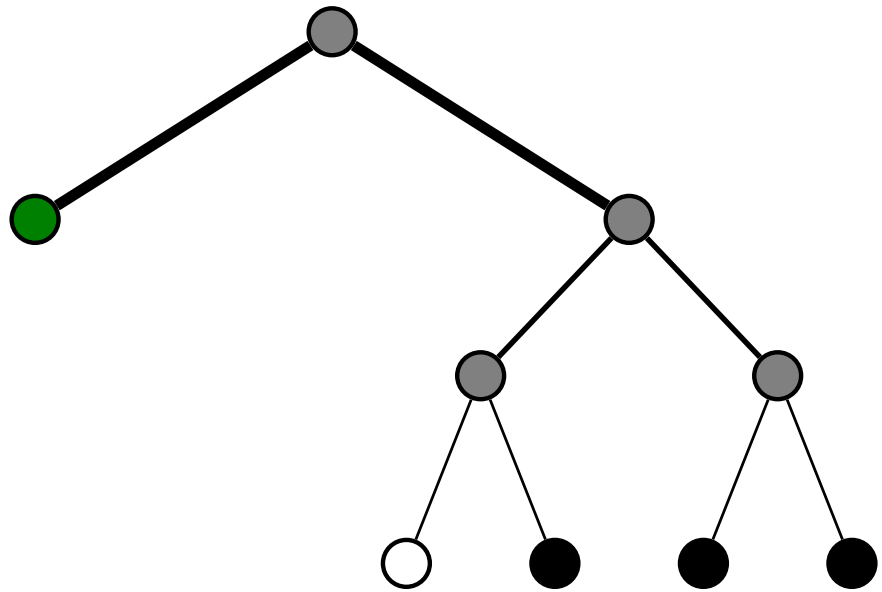
Fast-SSC Decoding



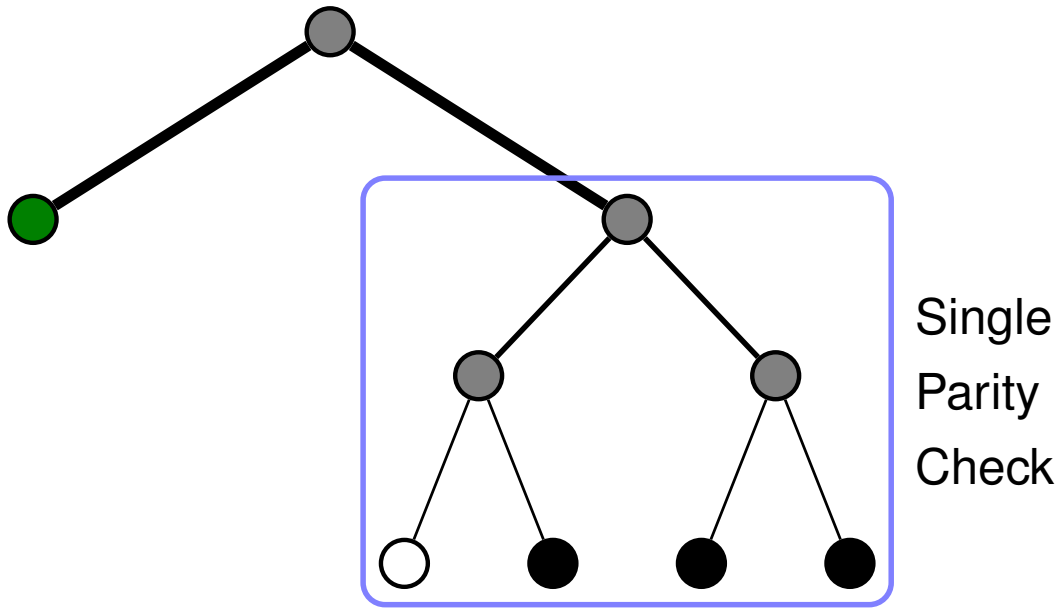
Fast-SSC Decoding



Fast-SSC Decoding

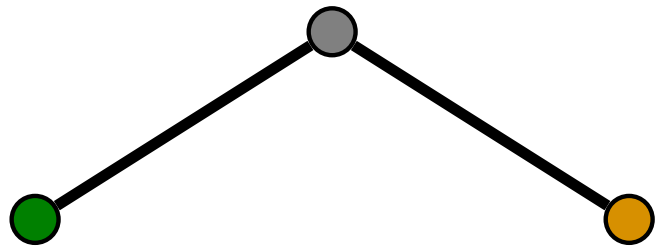


Fast-SSC Decoding

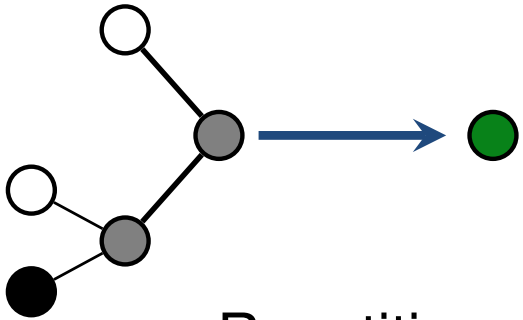


Sarkis et al., “Fast Polar Decoders: Algorithm and Implementation,” IEEE JSAC, 2014.

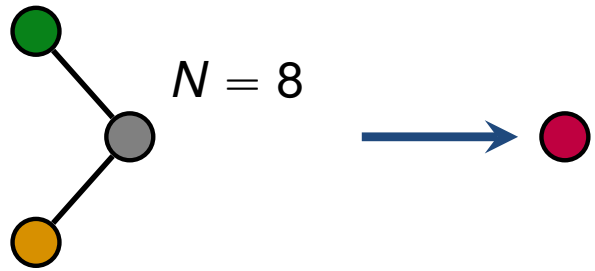
Fast-SSC Decoding



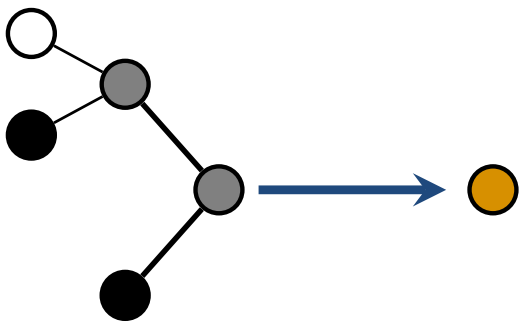
Fast-SSC Decoding—New Nodes



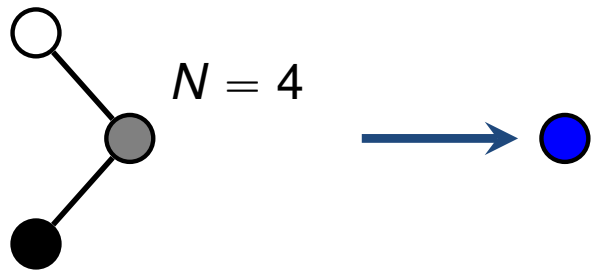
Repetition



Rep-SPC

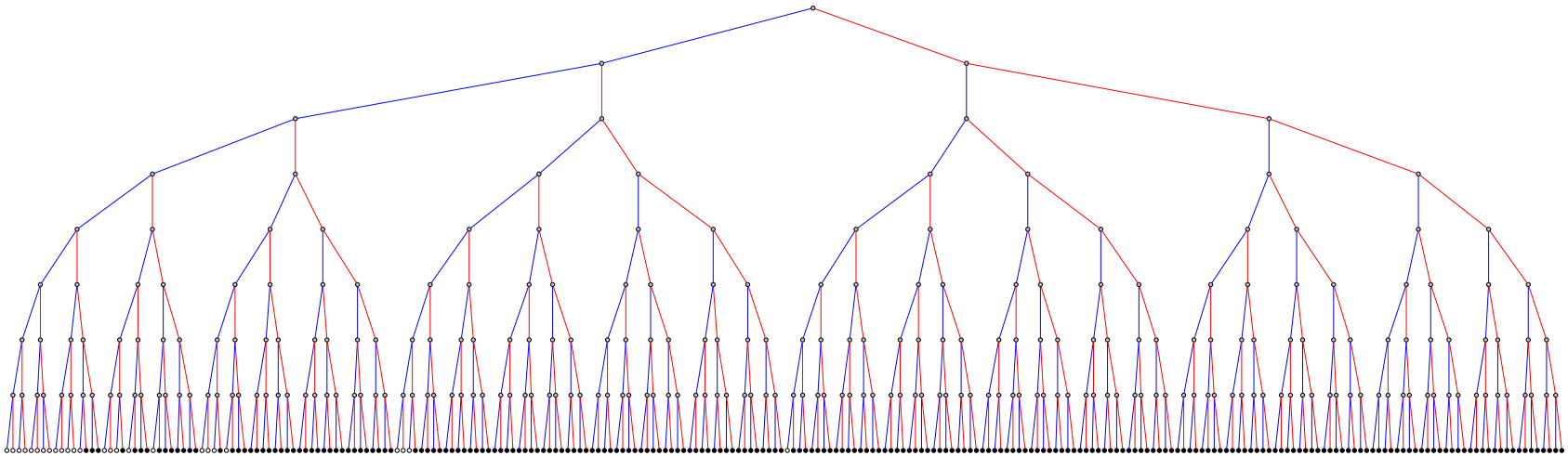


SPC

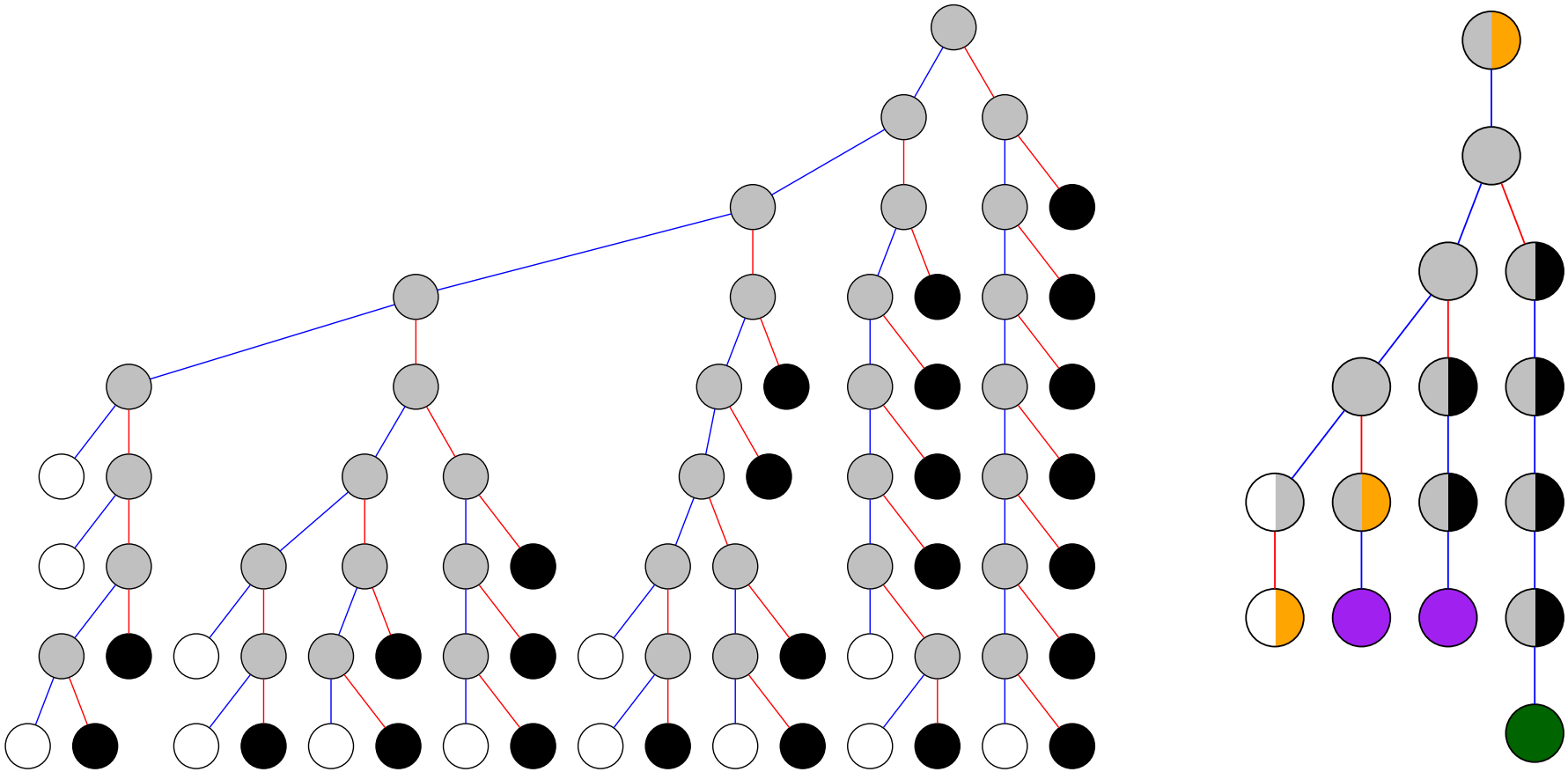


ML-0011

SC Tree—(256, 230)

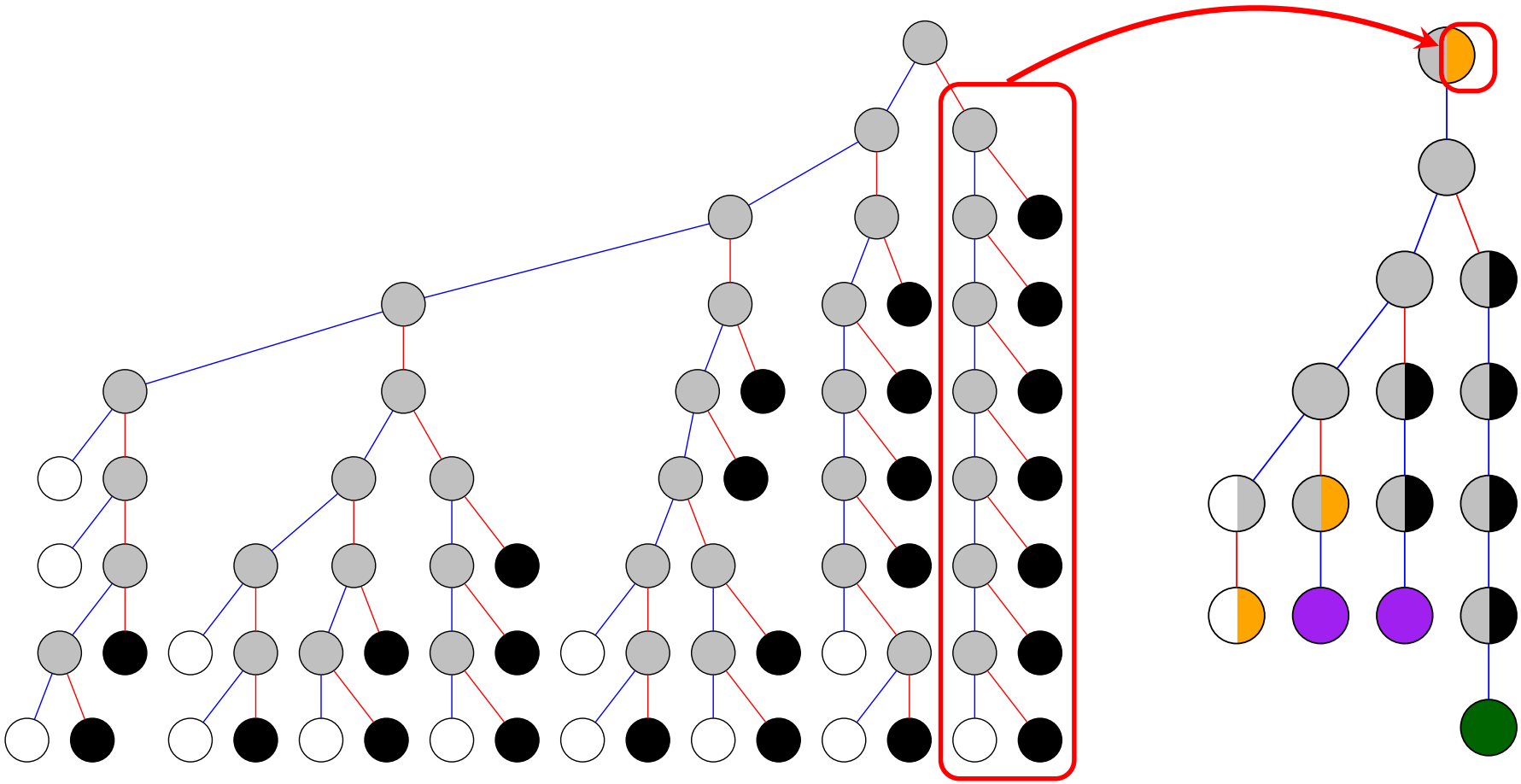


SSC and Fast-SSC Trees—(256, 230)



Developed a compiler to convert the graph to instructions.

SSC and Fast-SSC Trees—(256, 230)



Developed a compiler to convert the graph to instructions.

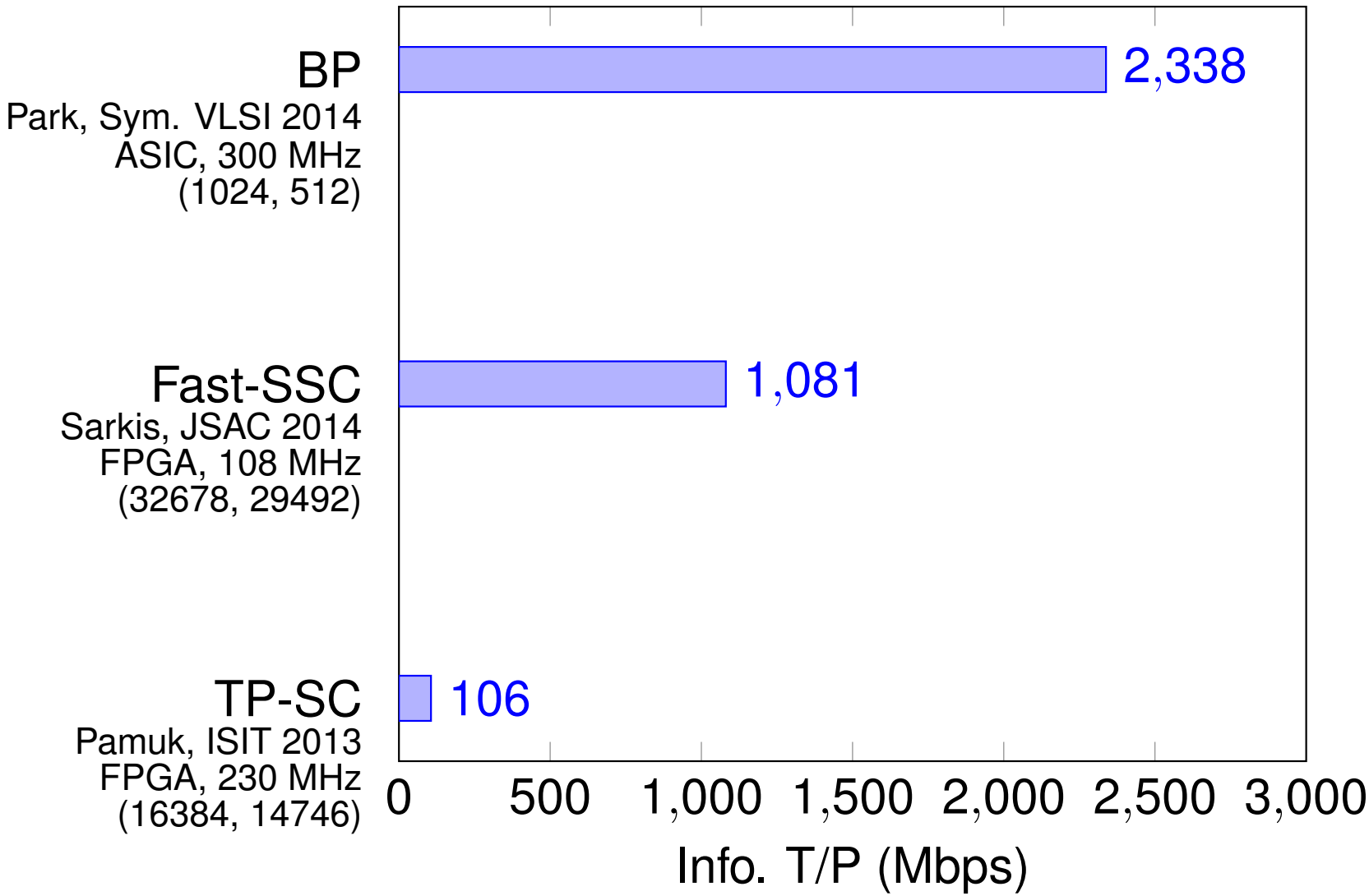
Fast-SSC Decoder Implementation

R	LUT	FF	RAM (bits)	f (MHz)	T/P (Mbps)
Fast-SSC					
0.83	25,866	7,209	536,126	108	791
0.90					1,081
SSC					
0.90					276

For any code of length = 32,768.

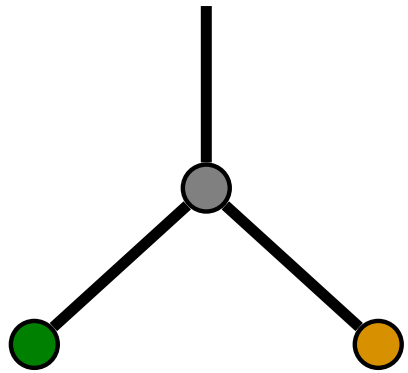
Sarkis et al. “Fast polar decoders: algorithm and implementation,” IEEE JSAC, 2014.

Throughput of Hardware Decoders



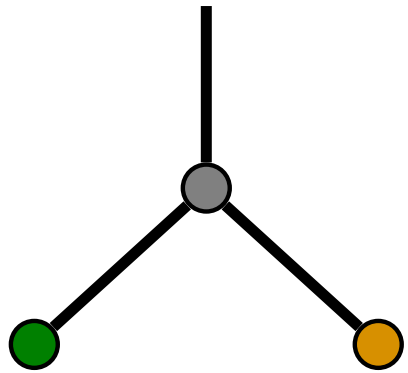
The Unrolled Decoder

- (8, 4) polar code:

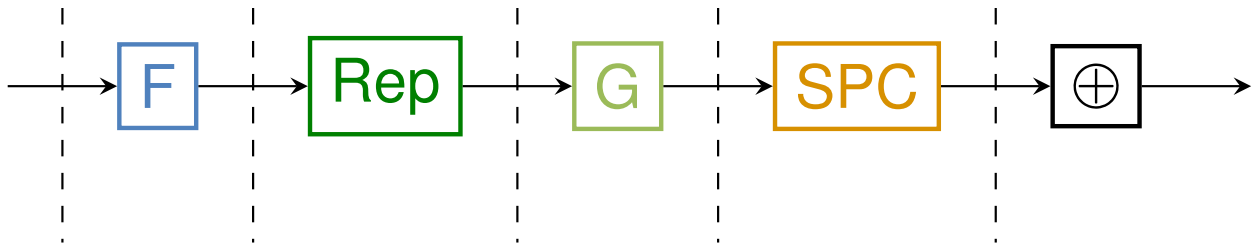


The Unrolled Decoder

- (8, 4) polar code:

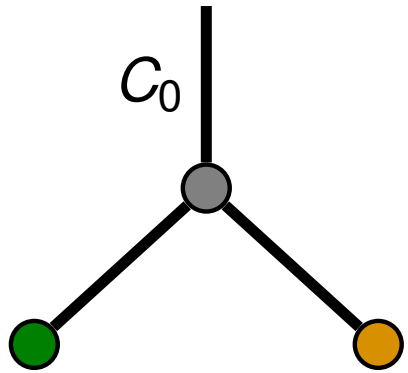


- Unroll and pipeline all operations.

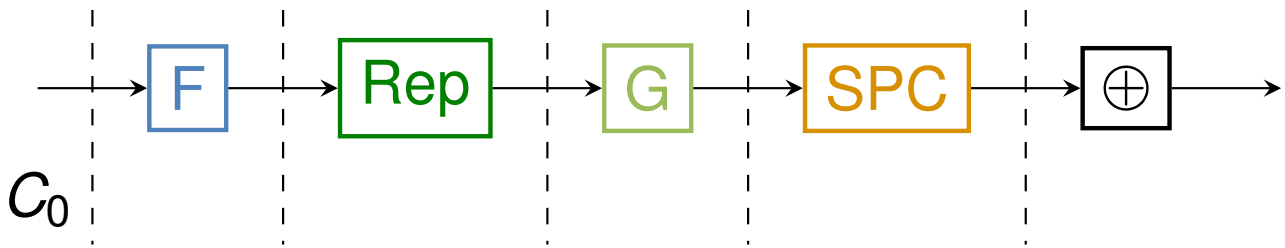


The Unrolled Decoder

- (8, 4) polar code:

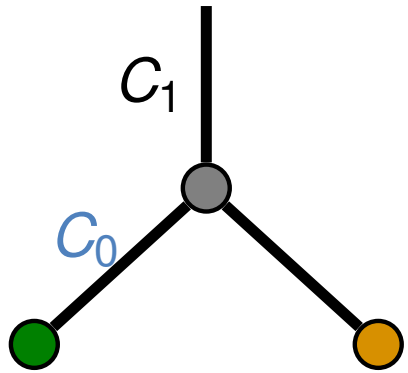


- Unroll and pipeline all operations.

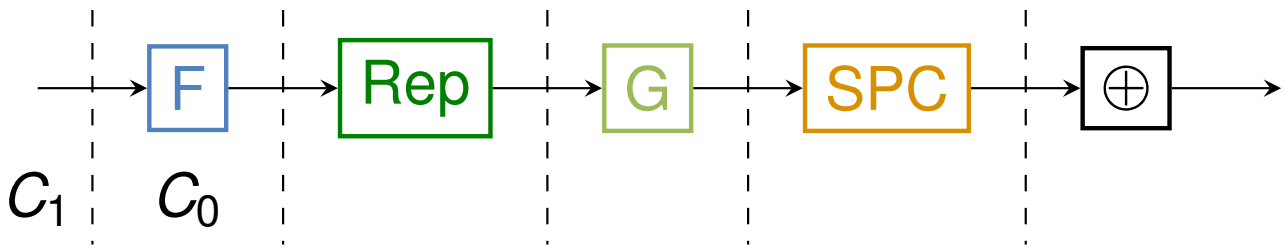


The Unrolled Decoder

- (8, 4) polar code:

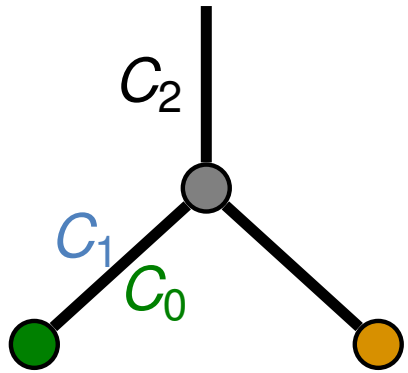


- Unroll and pipeline all operations.

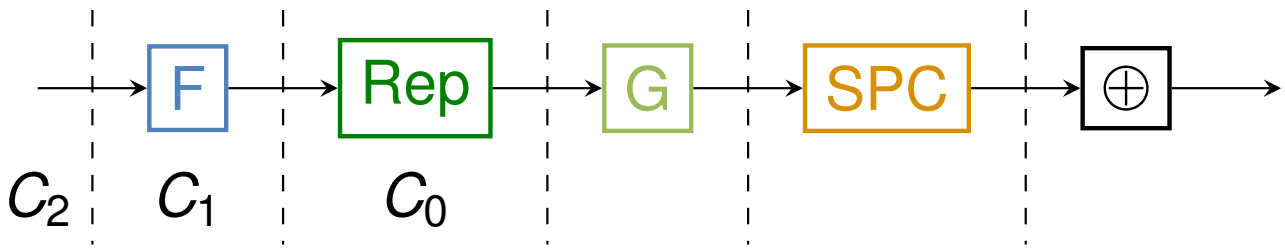


The Unrolled Decoder

- (8, 4) polar code:

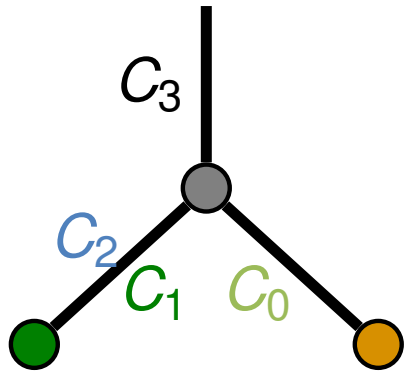


- Unroll and pipeline all operations.

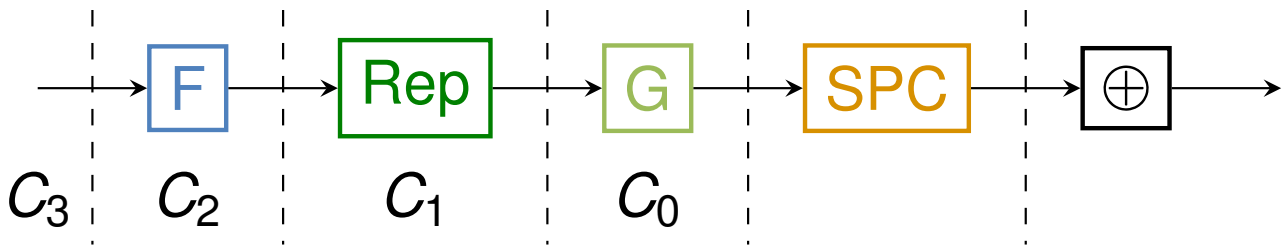


The Unrolled Decoder

- (8, 4) polar code:

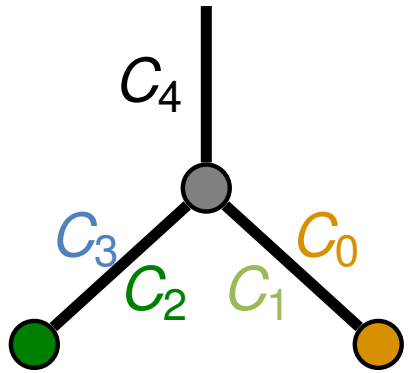


- Unroll and pipeline all operations.

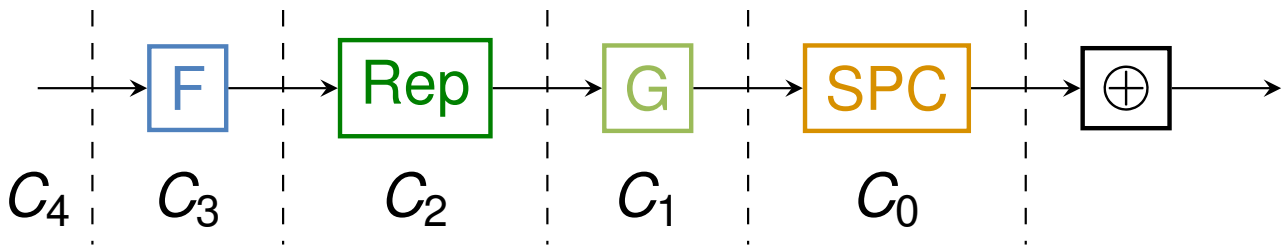


The Unrolled Decoder

- (8, 4) polar code:

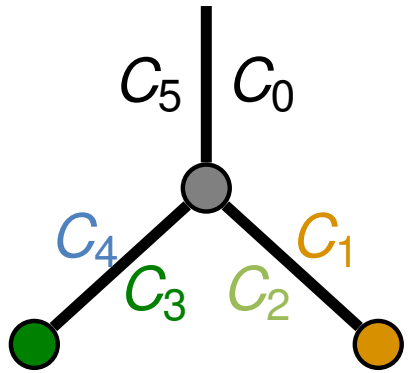


- Unroll and pipeline all operations.

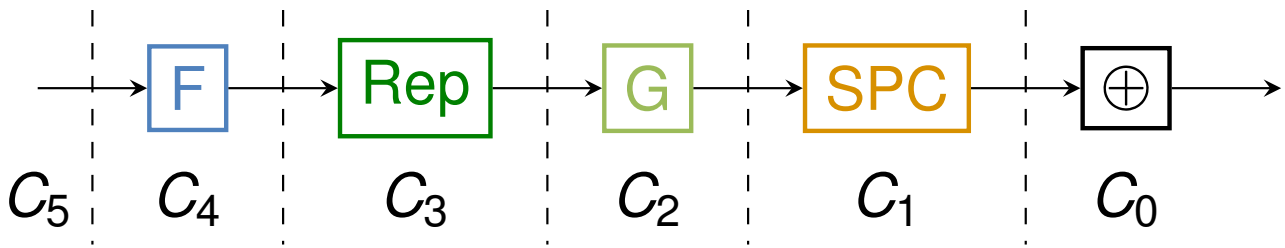


The Unrolled Decoder

- (8, 4) polar code:



- Unroll and pipeline all operations.

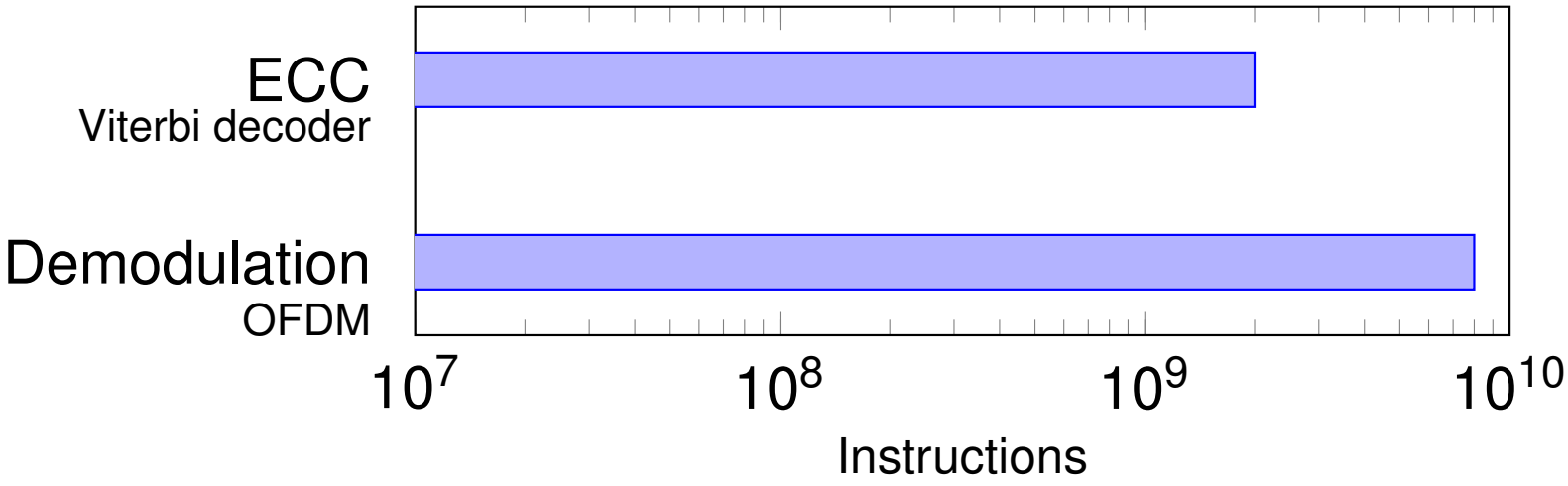


The Unrolled Decoder Implementation

LUT	FF	RAM (bits)	f (MHz)	Info. T/P (Gbps)
SP-SC				
4,130	1,388	11,904	196	0.04
Unrolled				
155,858	158,185	285,120	231	118.5
BP (ASIC)				
			300	2.3

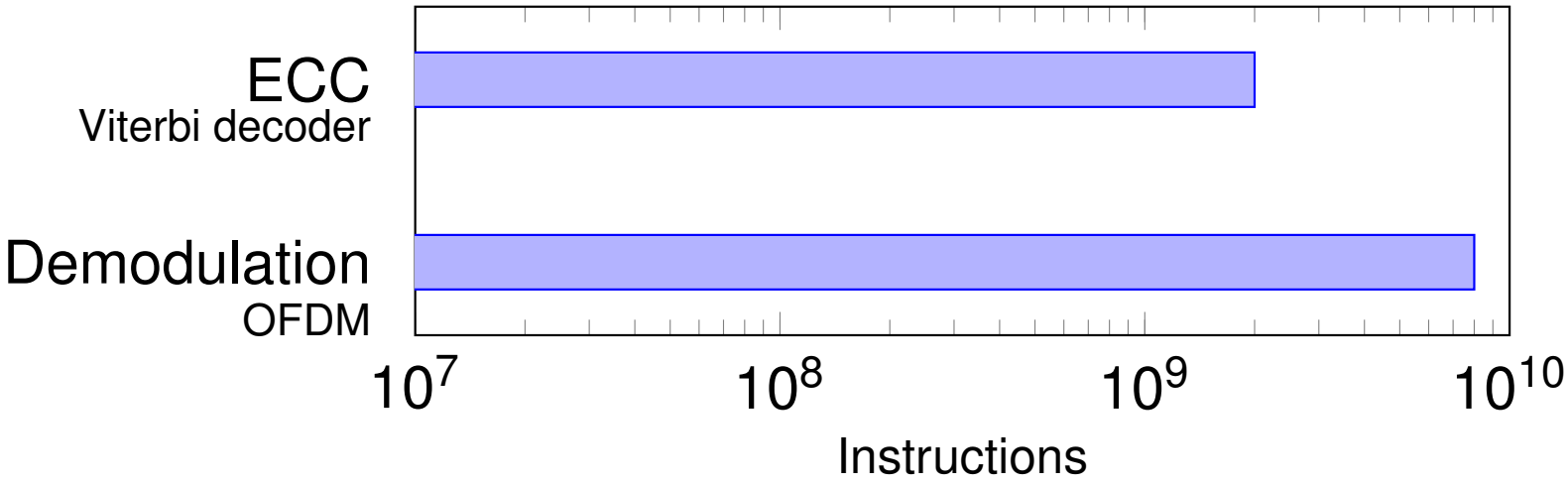
For a (1024, 512) polar code.

SDR and Modern Software ECC

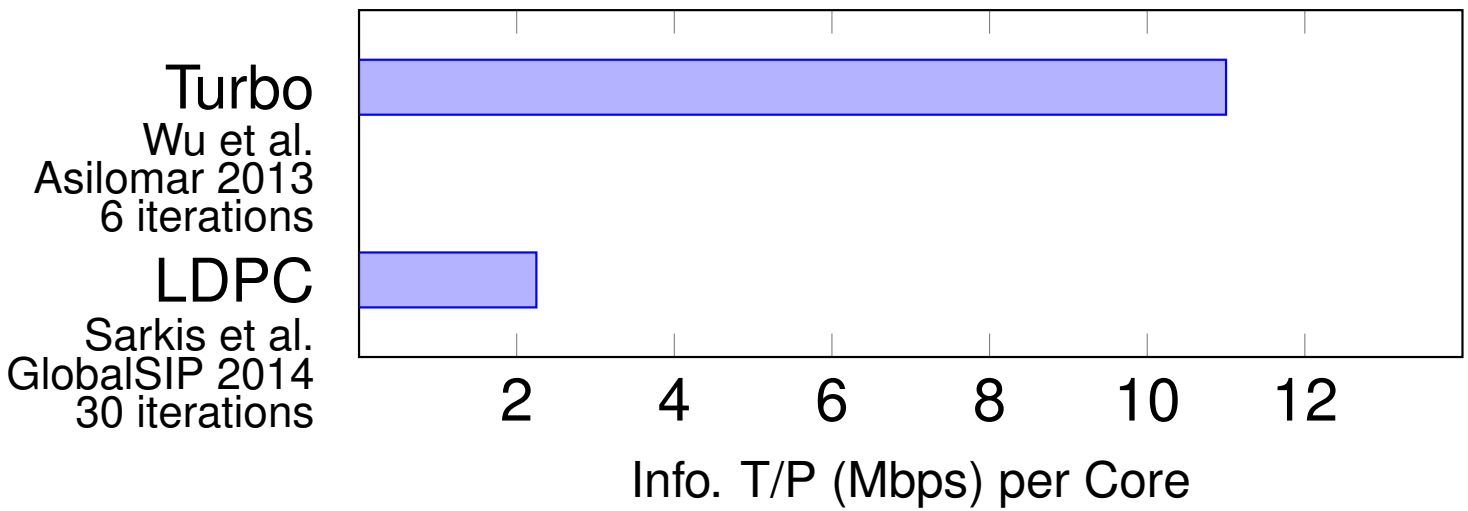


Demel et al. “A LTE Receiver Framework Using GNU Radio,” JSPS, 2015.

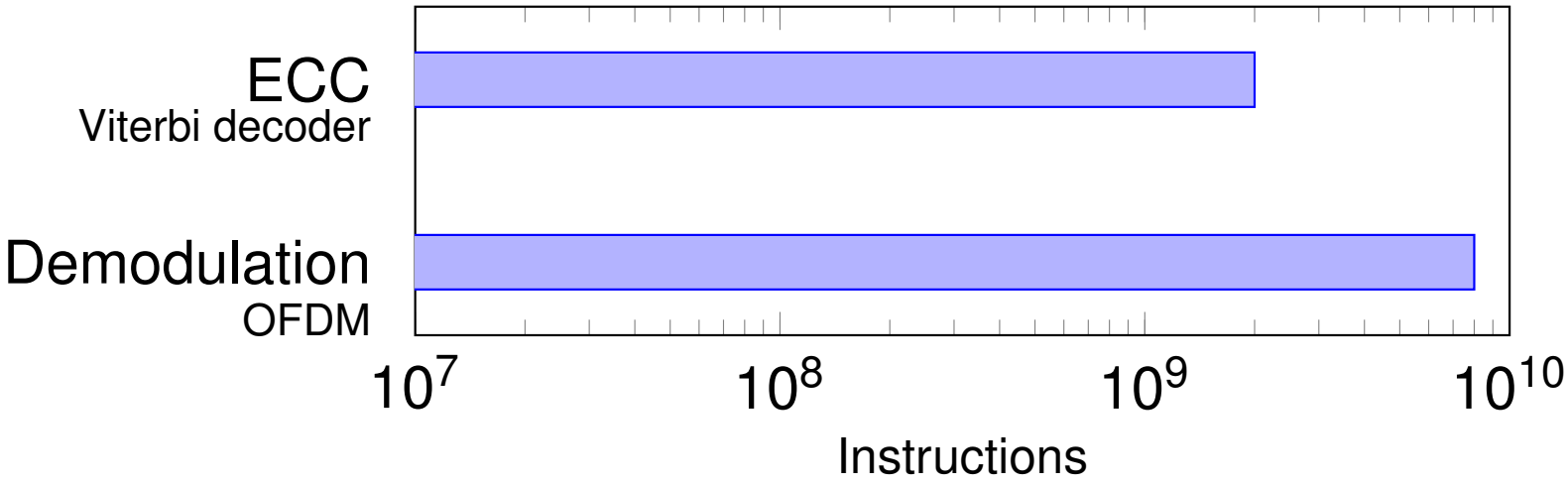
SDR and Modern Software ECC



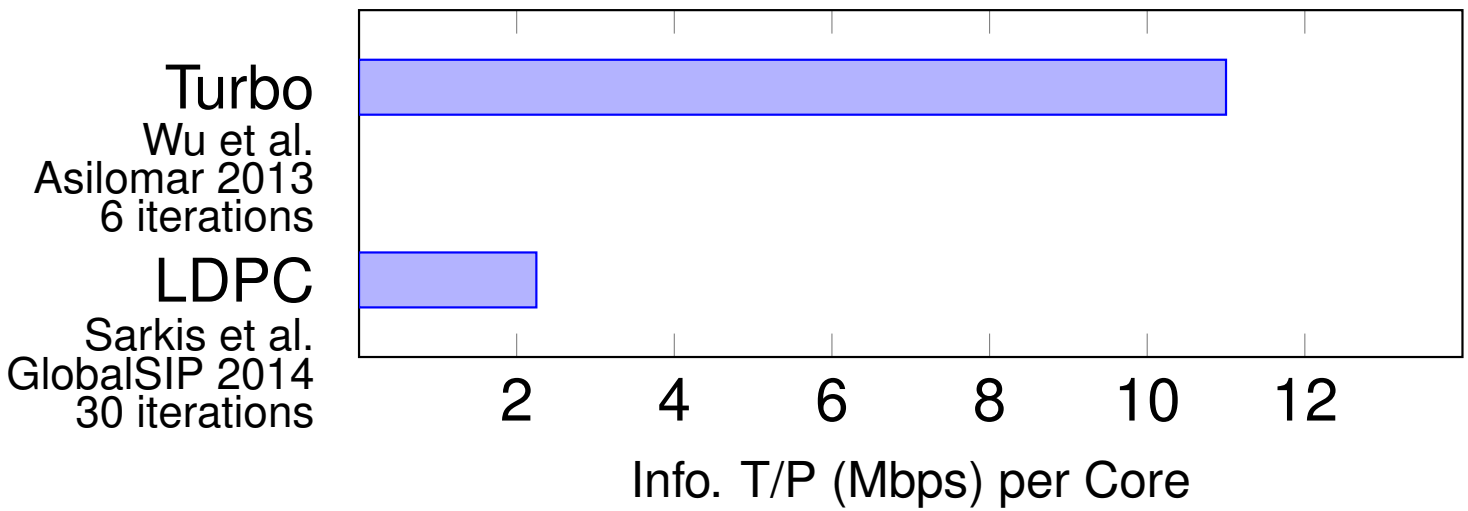
Demel et al. “A LTE Receiver Framework Using GNU Radio,” JSPS, 2015.



SDR and Modern Software ECC

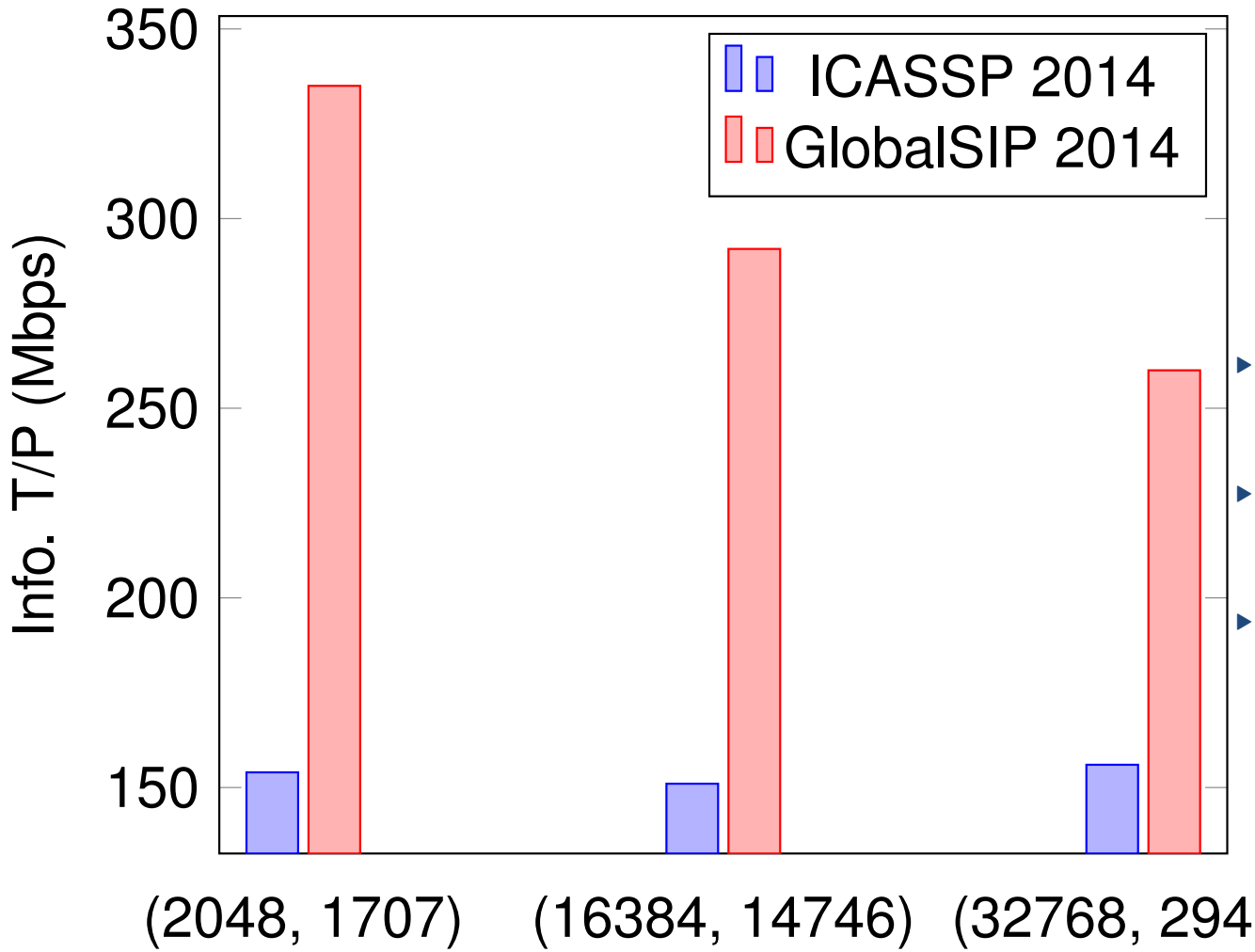


Demel et al. “A LTE Receiver Framework Using GNU Radio,” JSPS, 2015.



Goal: 50Mbps!

Floating-Point Throughput



- SIMD.
- Branchless.
- Single core.

Giard et al. “Fast Software Polar Decoders,” ICASSP 2014.
Sarkis et al. “Autogenerating Software Polar Decoders,” GlobalSIP 2014.

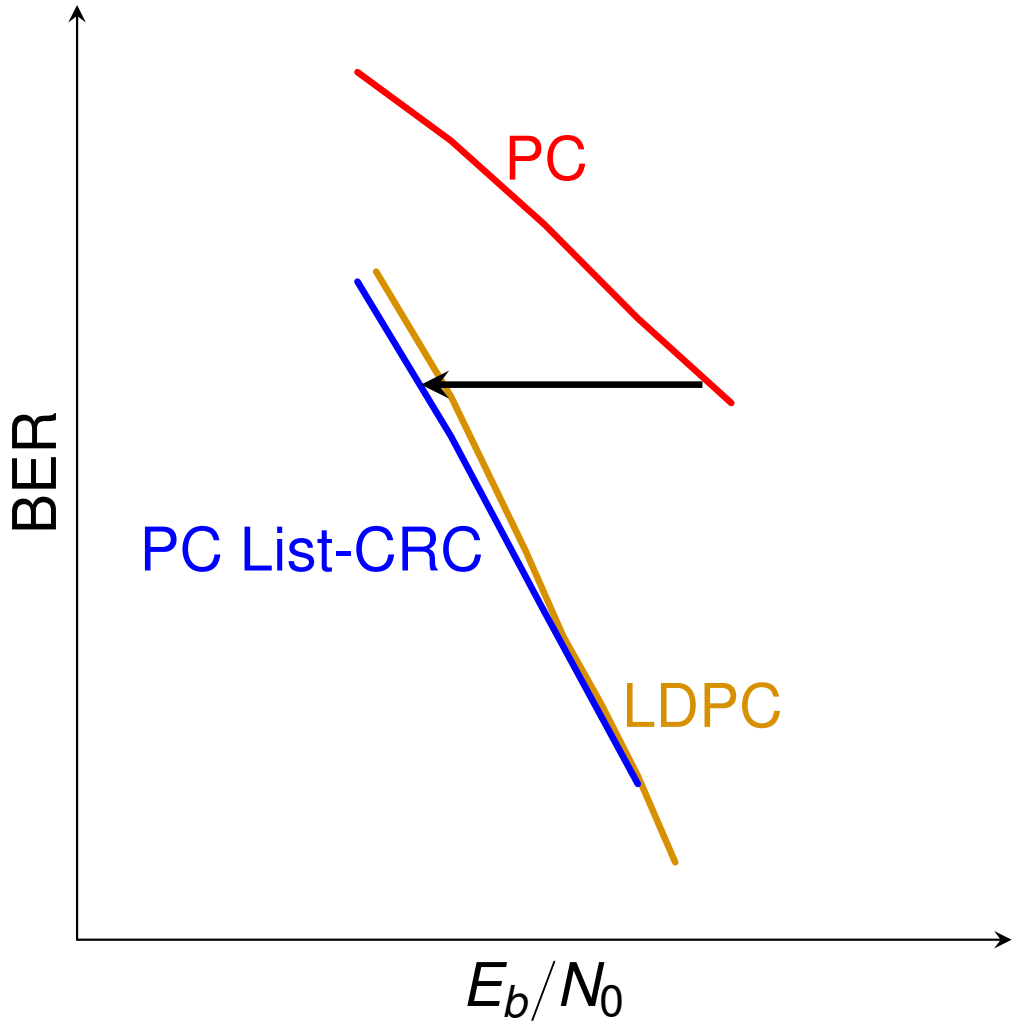
Fixed-Point Software Polar Decoders

Decoder	$f(GHz)$	Info. T/P (Mbps)	Latency μs
Le Gal et al.	3.6+	1,557	605
Fast-SSC-`int8`	3.1+	1,412	21

(32768, 29492) polar code.

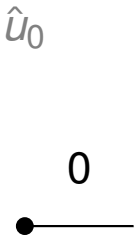
Le Gal et al. “Multi-Gb/s Software decoding of Polar Codes,” TSP 2015.
Giard et al. “Low-latency Software Polar Decoders,” *in preparation*.

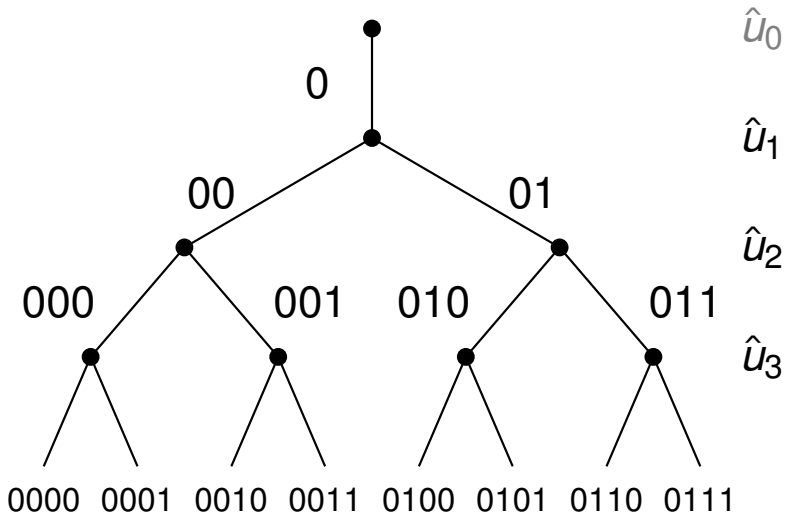
Improving Finite-Length Error-Correction Performance



List Decoding

- For each information bit, continue decoding assuming both 0 and 1.

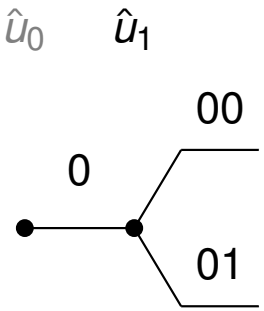


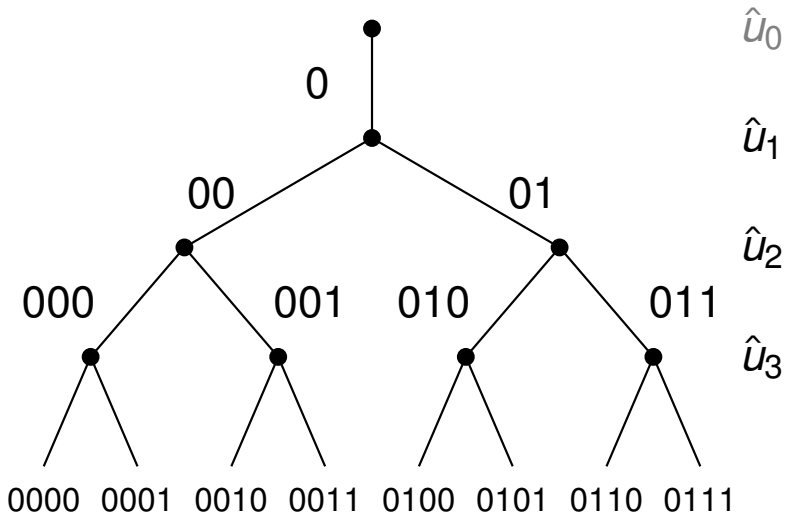


- Select paths according to reliability values.

List Decoding

- For each information bit, continue decoding assuming both 0 and 1.

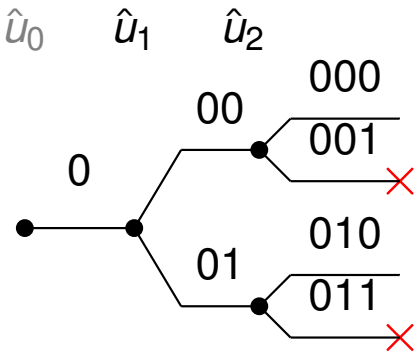


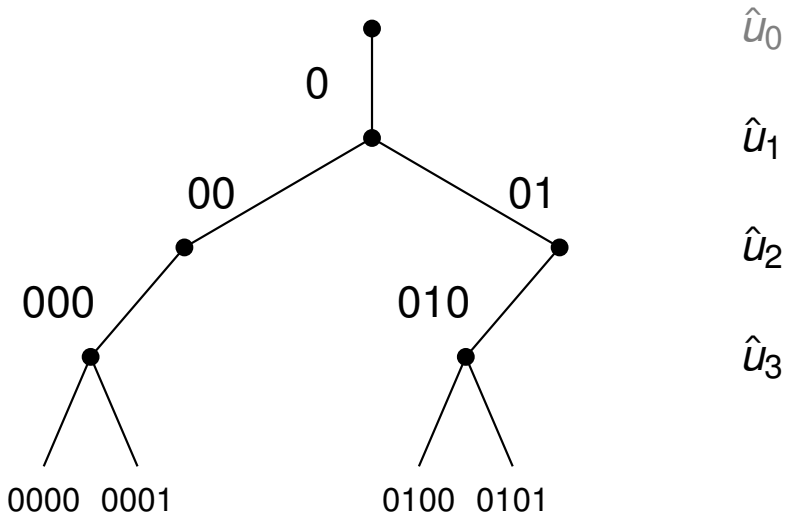


- Select paths according to reliability values.

List Decoding

- For each information bit, continue decoding assuming both 0 and 1.

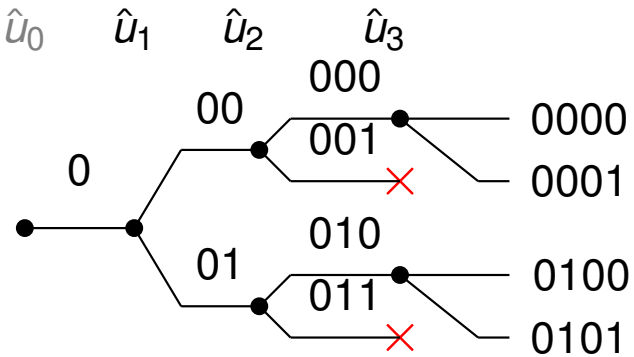


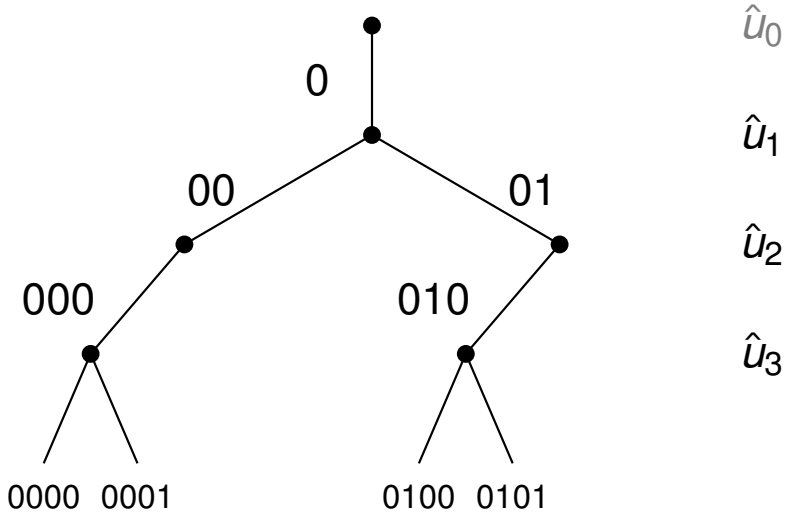


- Select paths according to reliability values.

List Decoding

- For each information bit, continue decoding assuming both 0 and 1.

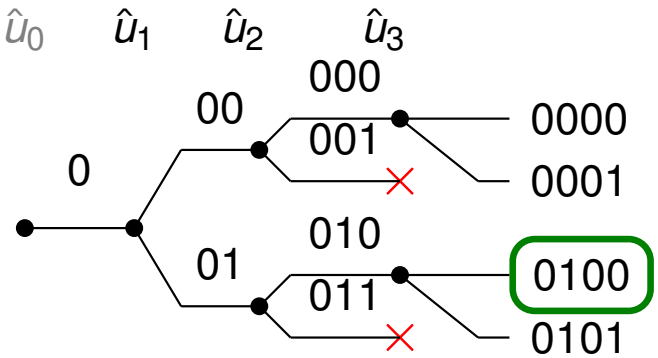


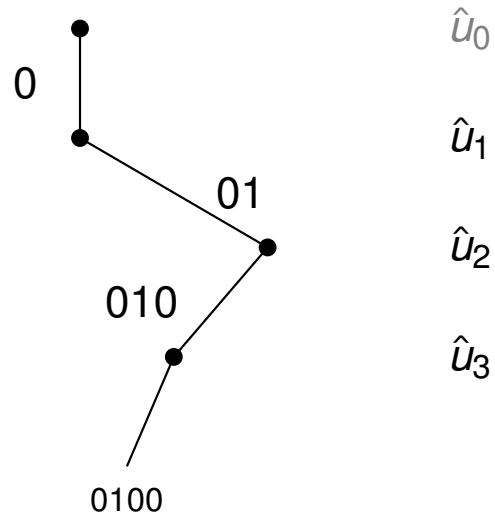


- Select paths according to reliability values.

List Decoding

- For each information bit, continue decoding assuming both 0 and 1.

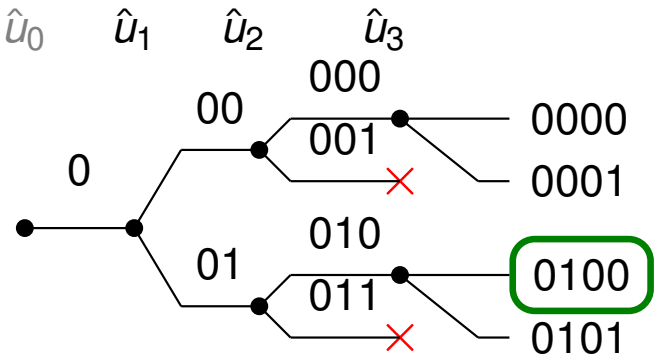


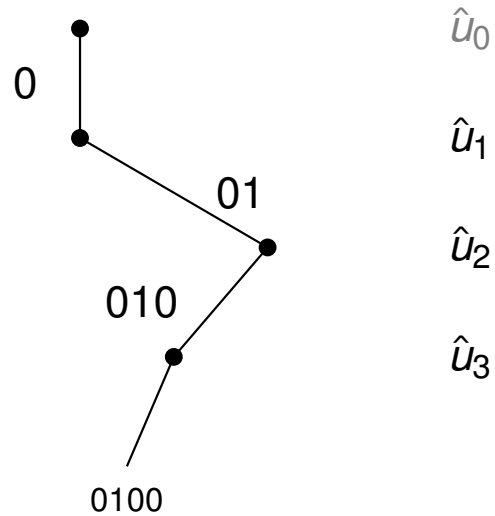


- Select paths according to reliability values.

List Decoding

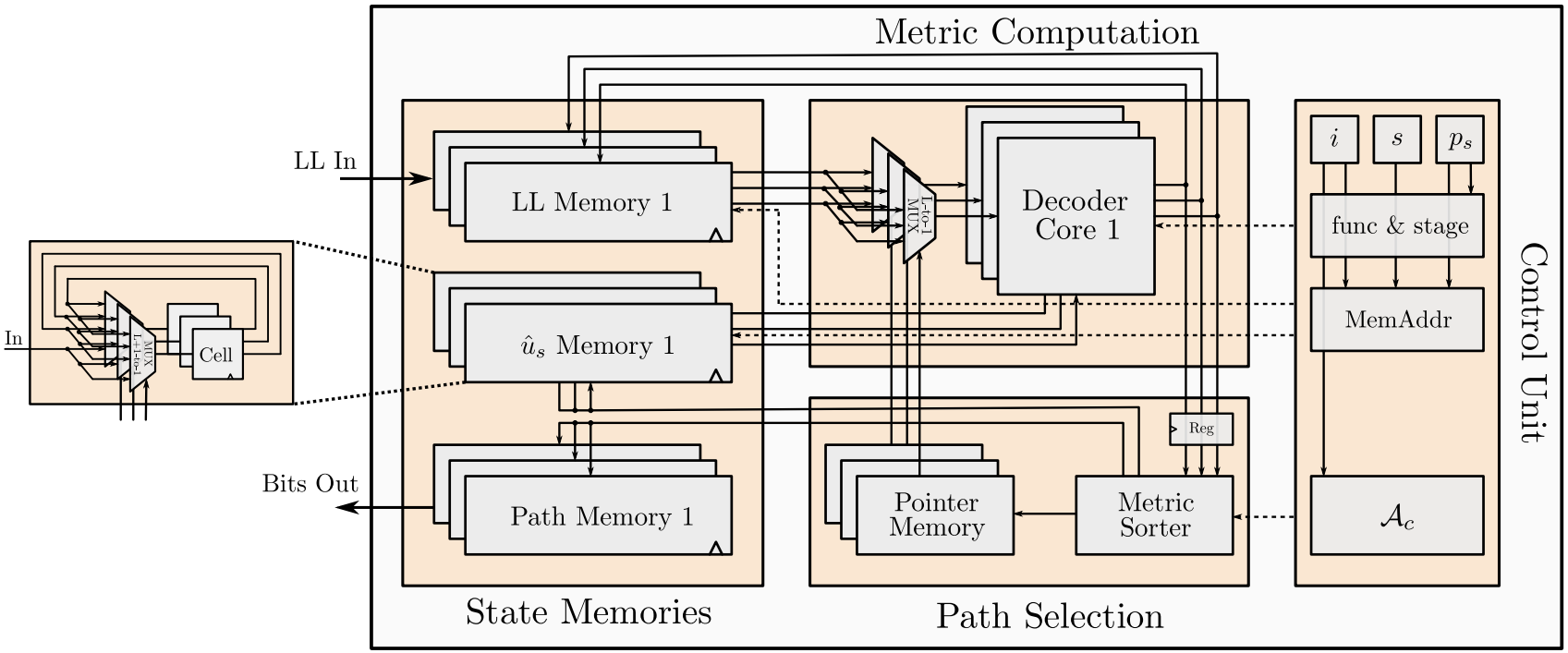
- For each information bit, continue decoding assuming both 0 and 1.





- Select paths according to reliability values.
- Using a CRC to select the output yields a significant improvement.

ASIC Implementation



- Synthesis results for $N = 1024$ using UMC 90nm:

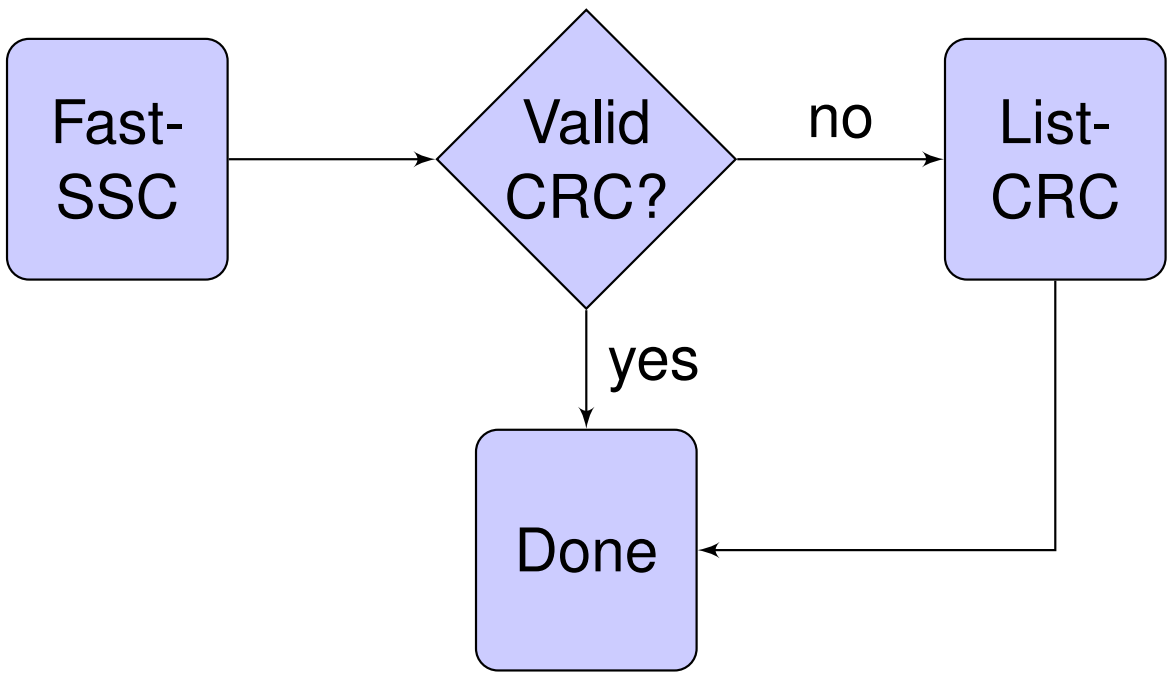
L	Area (mm^2)	f (MHz)	T/P (Mbps)
2	1.60	459	$181R$
4	3.53	314	$89R$

Two-Step SW List Decoder

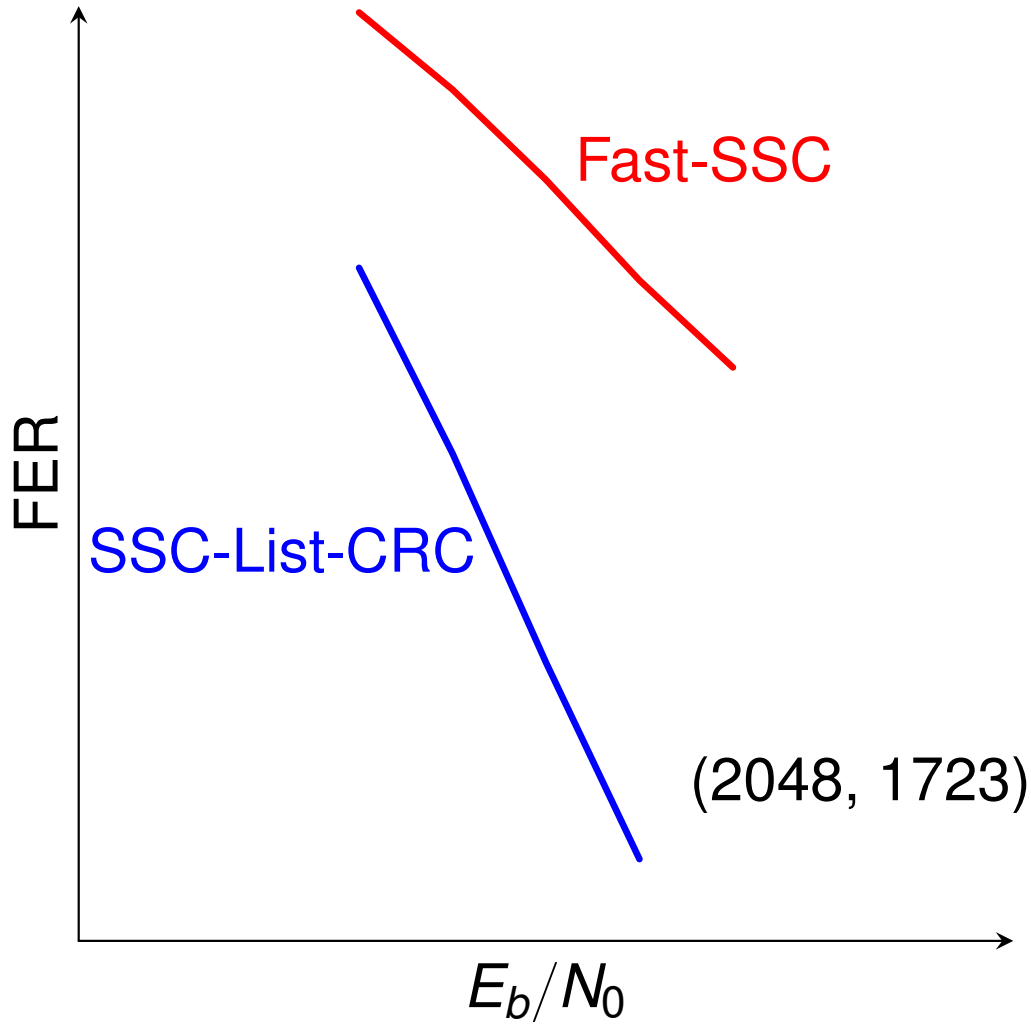
- List decoders are slower than Fast-SSC decoders.
- List have better error-correction performance than Fast-SSC decoders.

Two-Step SW List Decoder

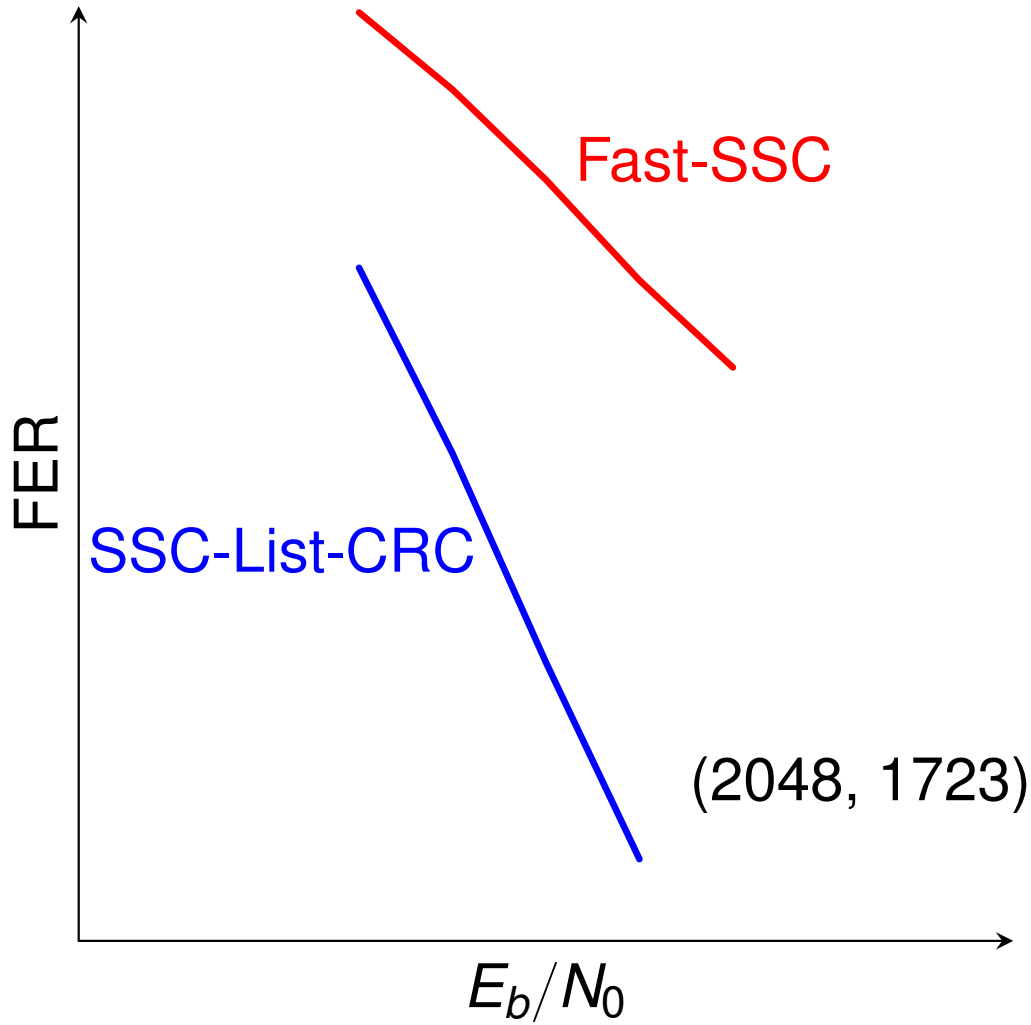
- List decoders are slower than Fast-SSC decoders.
- List have better error-correction performance than Fast-SSC decoders.
- Combine both:



Two-Step SW List Decoder

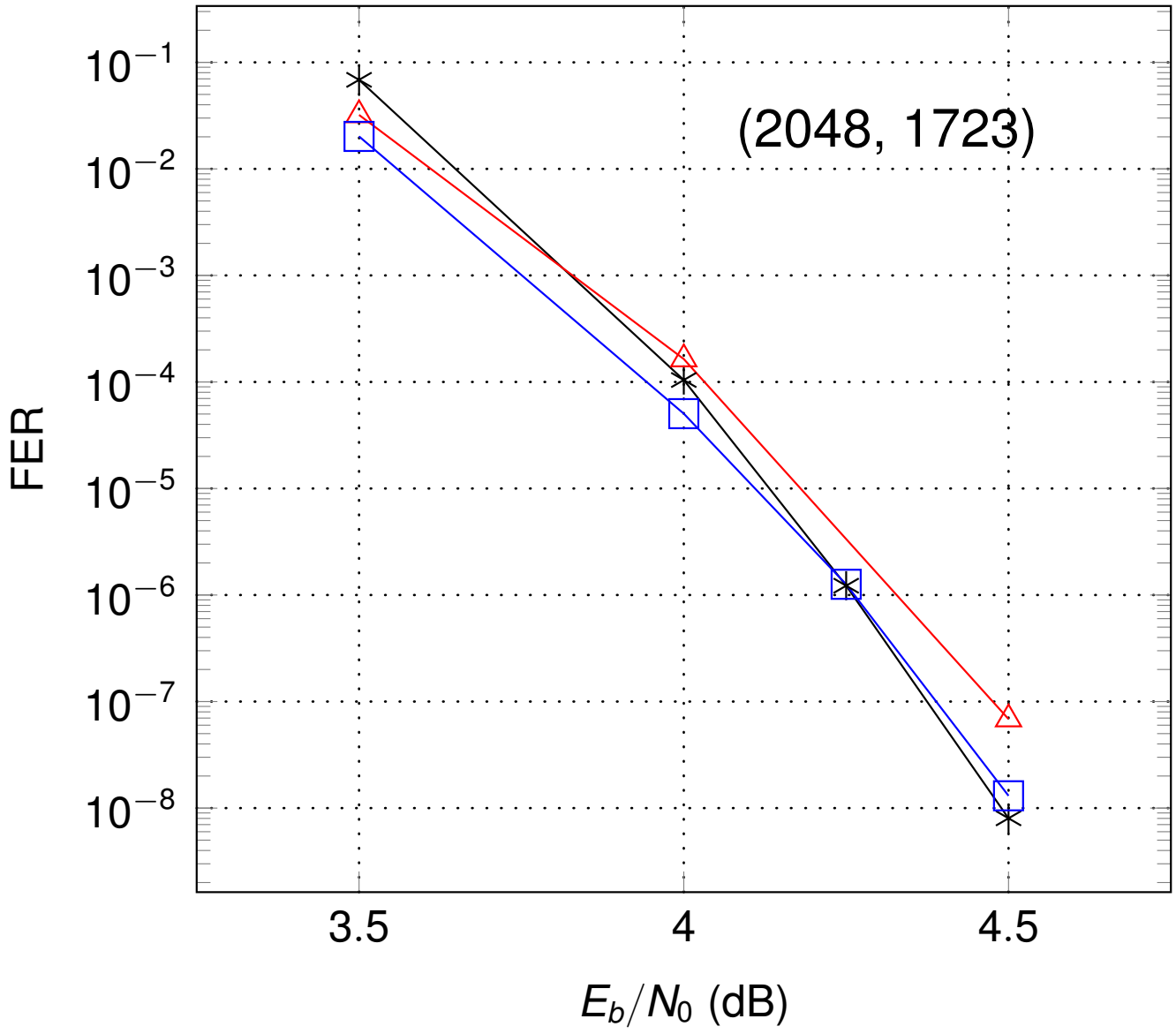


Two-Step SW List Decoder

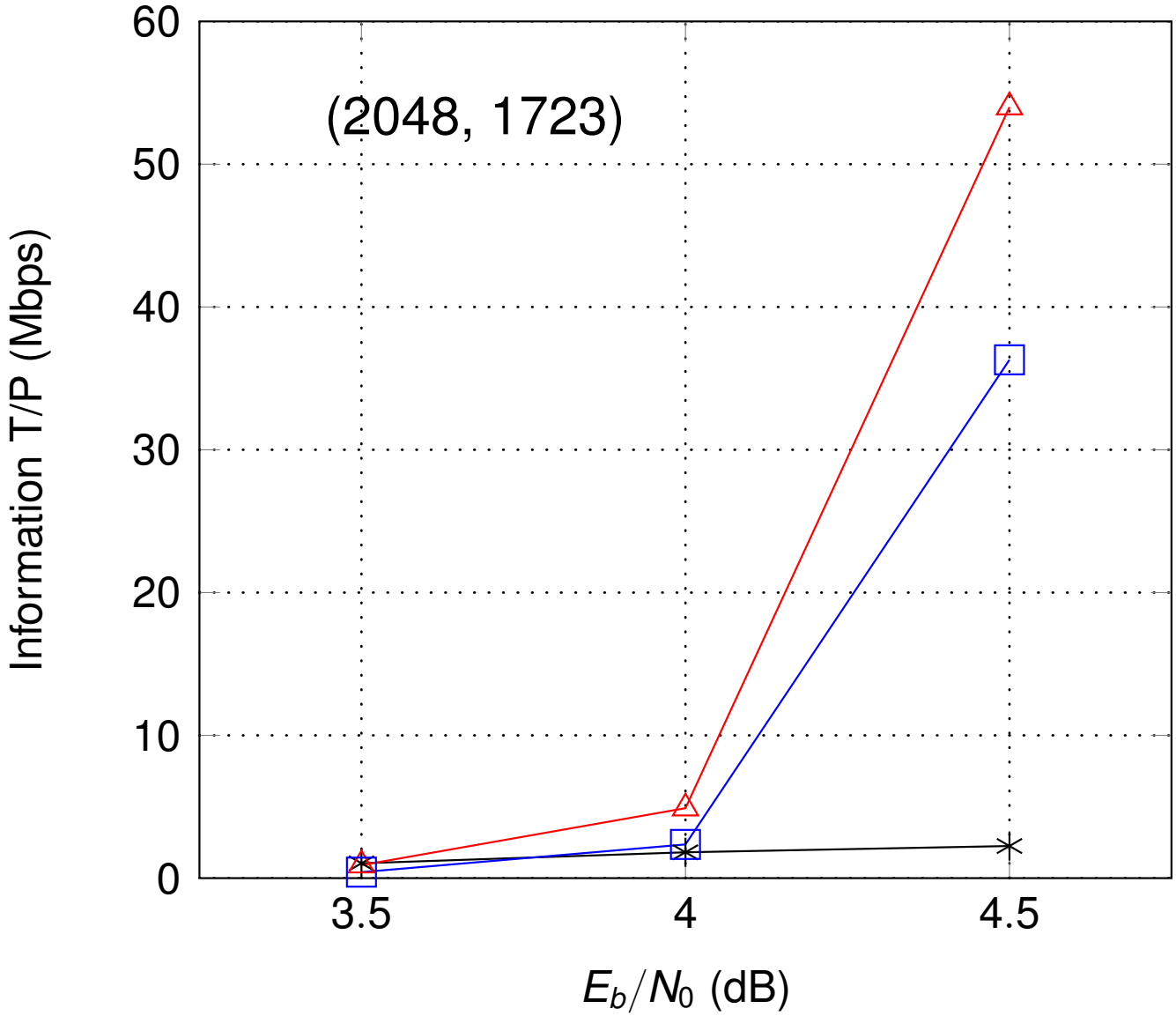


$$\mathcal{T} = \frac{k}{(1-\mathrm{FER_F})\mathcal{L}(\text{Fast-SSC}) + \mathrm{FER_F}\mathcal{L}(\text{List})} \text{ bit/s}$$

Two-Step SW List Decoder



Two-Step SW List Decoder



Conclusion

- Polar codes can have excellent error-correction performance.
- Polar codes can have throughput in the hundred-Gbps range.
- Once throughput and performance improvements are combined, we believe that polar codes will be a viable option for multiple applications.

Conclusion

- Polar codes can have excellent error-correction performance.
- Polar codes can have throughput in the hundred-Gbps range.
- Once throughput and performance improvements are combined, we believe that polar codes will be a viable option for multiple applications.

Thank you!