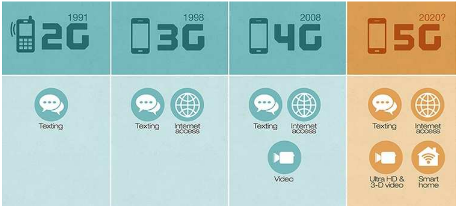
# Decoding Reed-Muller and Polar Codes by Successive Factor Graph Permutations

**Seyyed Ali Hashemi**[a], Nghia Doan[a], Marco Mondelli[b], Warren J. Gross[a]
[a]McGill University, Canada
[b]Stanford University, USA

ISTC 2018
Hong Kong
December 4, 2018.

# Motivation




- **Polar Codes**: adopted in **5G** eMBB control channel
  - Requires low-complexity decoders with good performance
- **Reed-Muller (RM) Codes**: very similar to polar codes

## In this talk:

We propose a new low-complexity decoder for RM and polar codes with good performance!

# RM and Polar Codes: Encoding

$$\mathcal{RM}(8,4),\ \mathcal{P}(8,4)$$

$$\mathbf{u}\mathbf{G}^{\otimes 3} = \mathbf{x}$$

# RM and Polar Codes: Encoding

$$\mathcal{RM}(8,4),\ \mathcal{P}(8,4)$$

$$\mathbf{u}\mathbf{G}^{\otimes 3} = \mathbf{x}$$

$$\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}^{\mathsf{T}}$$

# RM and Polar Codes: Encoding

$$\mathcal{RM}(8,4),\ \mathcal{P}(8,4)$$

$$\mathbf{u}\mathbf{G}^{\otimes 3} = \mathbf{x}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ u_3 \\ 0 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}^{\mathsf{T}}$$

# RM and Polar Codes: Encoding

$$\mathcal{RM}(8,4),\ \mathcal{P}(8,4)$$

$$\mathbf{uG}^{\otimes 3} = \mathbf{x}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ u_3 \\ 0 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}^{\mathsf{T}}$$

**RM rule**:

Remove rows with lowest Hamming weights

# RM and Polar Codes: Encoding

$$\mathcal{RM}(8,4),\ \mathcal{P}(8,4)$$

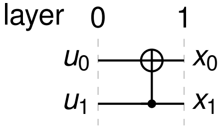
$$\mathbf{u}\mathbf{G}^{\otimes 3} = \mathbf{x}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ u_3 \\ 0 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}^{\mathsf{T}}$$

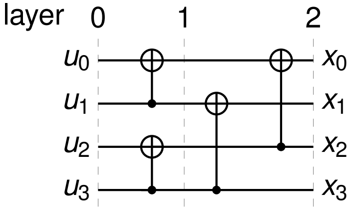
**Polar rule**:

Remove rows with lowest reliabilities

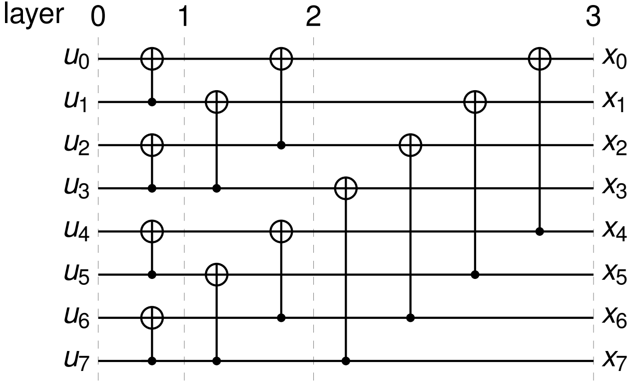
# RM and Polar Codes: Recursive Construction



# RM and Polar Codes: Recursive Construction

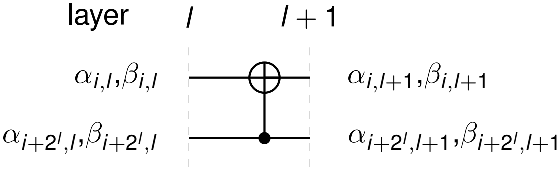


# RM and Polar Codes: Recursive Construction



# RM and Polar Codes: Decoding

Successive-Cancellation (SC)




$$\alpha_{i,l} = f\left(\alpha_{i,l+1}, \alpha_{i+2^l,l+1}\right)$$
$$\alpha_{i+2^l,l} = g\left(\alpha_{i,l+1}, \alpha_{i+2^l,l+1}, \beta_{i,l}\right)$$
$$\beta_{i,l+1} = \beta_{i,l} \oplus \beta_{i+2^l,l}$$
$$\beta_{i+2^l,l+1} = \beta_{i+2^l,l}$$

# RM vs Polar

- RM:
  - Minimizes error probability under MAP decoding
    - High complexity
  - Channel-independent
    - Low complexity
- Polar:
  - Minimizes error probability under SC decoding
    - Low complexity
  - Channel-dependent
    - High complexity

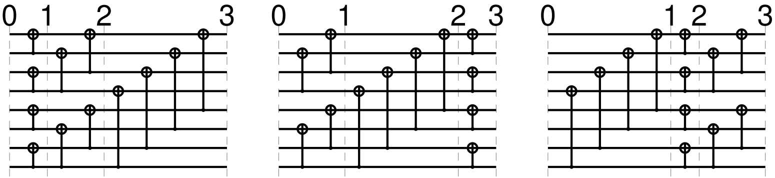
# SC List (SCL) Decoding: Something in Between

- SCL instantiates multiple SC decoders
  - $L$ codeword candidates survive to limit complexity
  - A CRC can help SCL find the correct candidate

$$\text{SC} \longleftrightarrow \text{SCL} \longleftrightarrow \text{MAP}$$

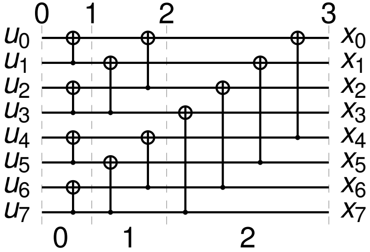
- Requires a large $L$ to get close to MAP

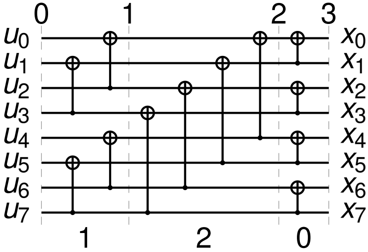
# Factor Graph Permutations

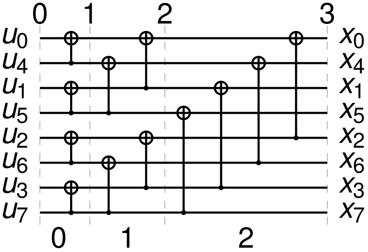



- $n = \log_2 N$ layers $\rightarrow$ $n!$ factor graph permutations
- Decode over multiple factor graphs $\rightarrow$ Pick the best one
  - High decoding complexity

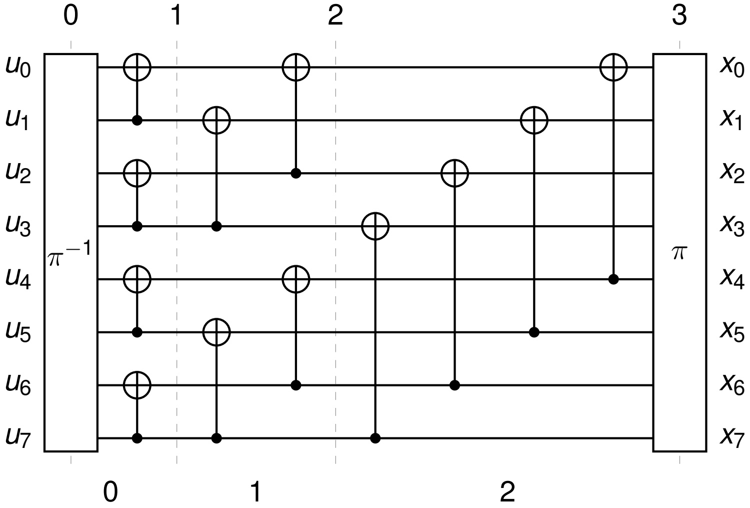
# Bit Index Permutations



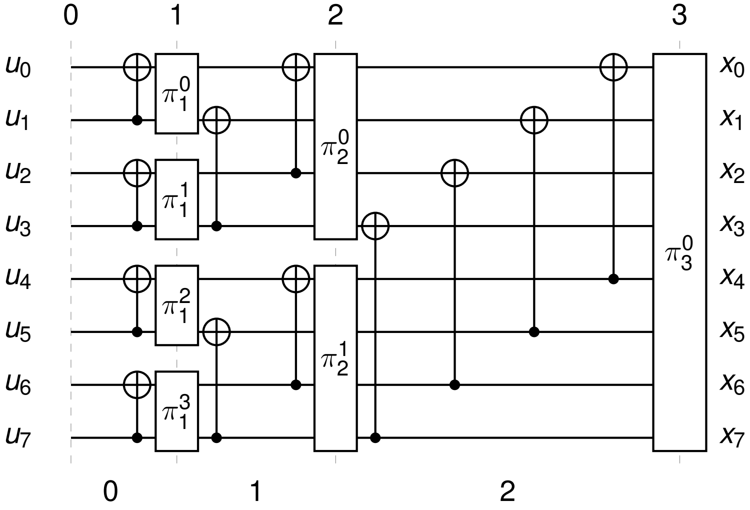



# Architecture



# Permutations on Sub-Graphs



# Total Number of Permutations

- By cyclic shift permutations at each layer:

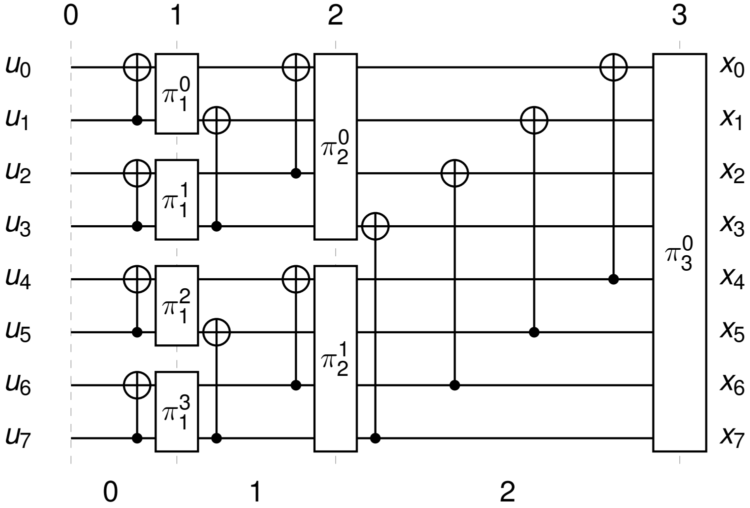
$$\prod_{l=0}^{n-1} (n-l)^{(2^l)}$$

- Much larger than $n$!
- Quickly grows with $n$
  - 1658880 permutations for length 32
  - More than $1.9 \times 10^{27}$ for length 128

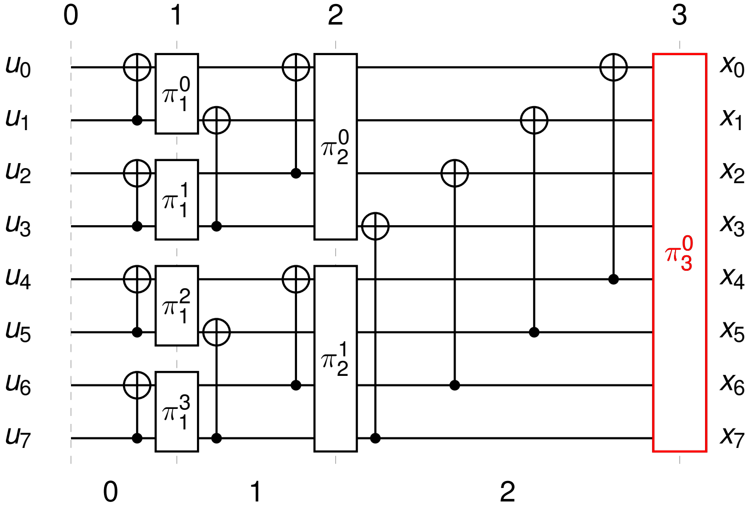
## Issue:

Decoding over all permutations is impossible!

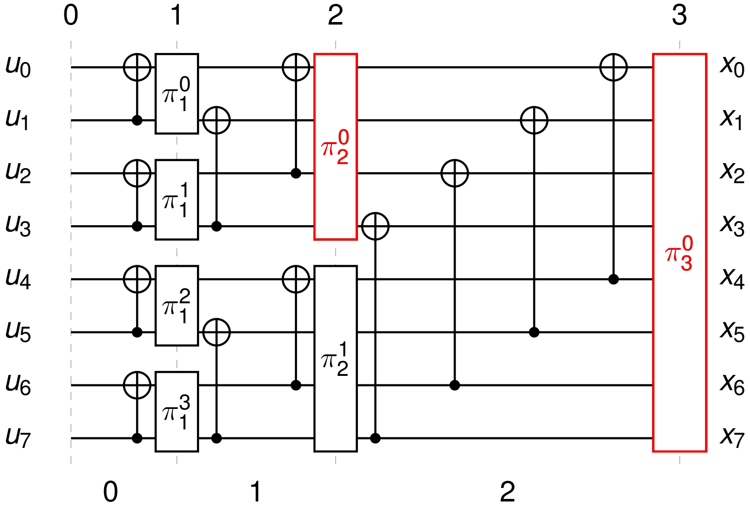
# Successive Permutations for SC



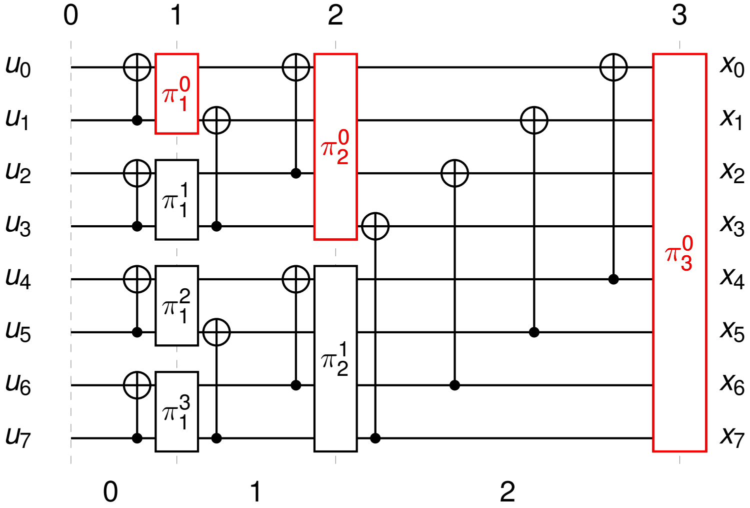
# Successive Permutations for SC



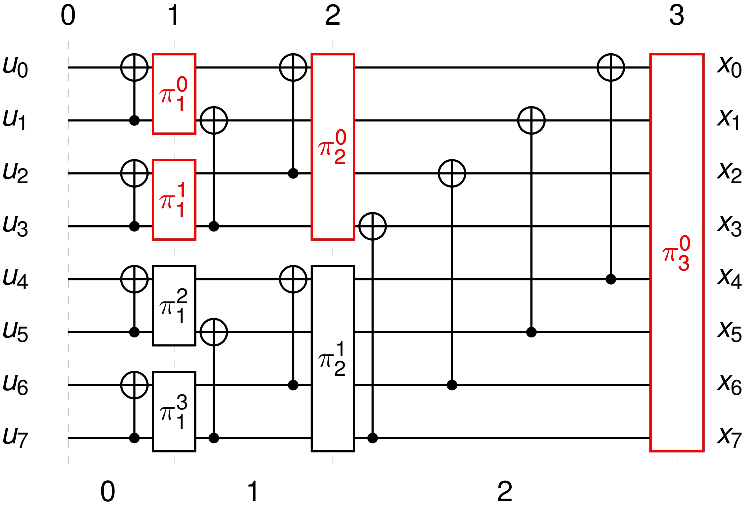
# Successive Permutations for SC



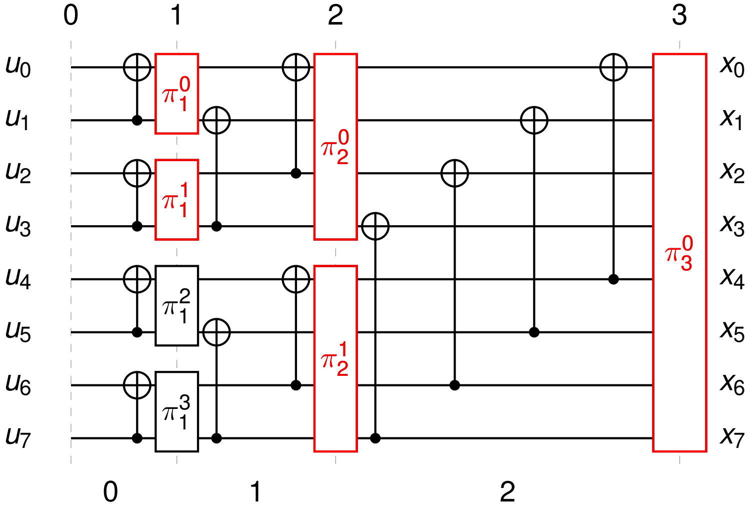
# Successive Permutations for SC



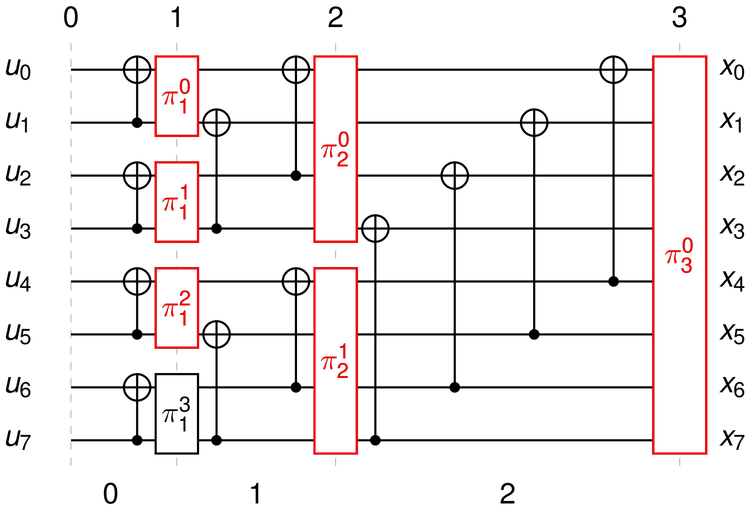
# Successive Permutations for SC



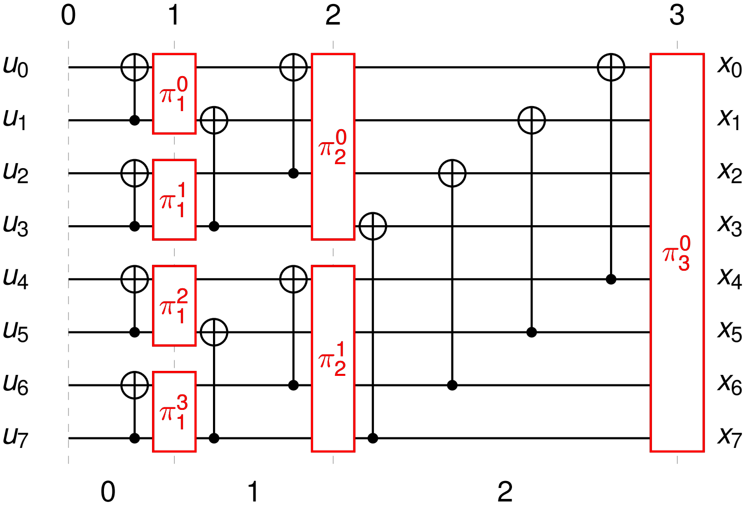
# Successive Permutations for SC



# Successive Permutations for SC



# Successive Permutations for SC



# Selection Criterion

- Reliability of a vector of LLR values $\boldsymbol{\alpha}$ of length $N$:

$$\mathcal{R}(\boldsymbol{\alpha}) = \sum_{i=0}^{N-1} |\alpha_i|$$

- A larger $\mathcal{R}(\boldsymbol{\alpha})$ indicates a more reliable LLR vector

## Criterion:

Permutation with which the $f$ function results in the most reliable LLR vector

$$P_{l+1} = \underset{p_{l+1} \in \pi_{l+1}}{\arg\max} \sum_{i=0}^{2^l-1} \left| f\left(\alpha_{p_{l+1}(i),l+1}, \alpha_{p_{l+1}(i+2^l),l+1}\right) \right|$$

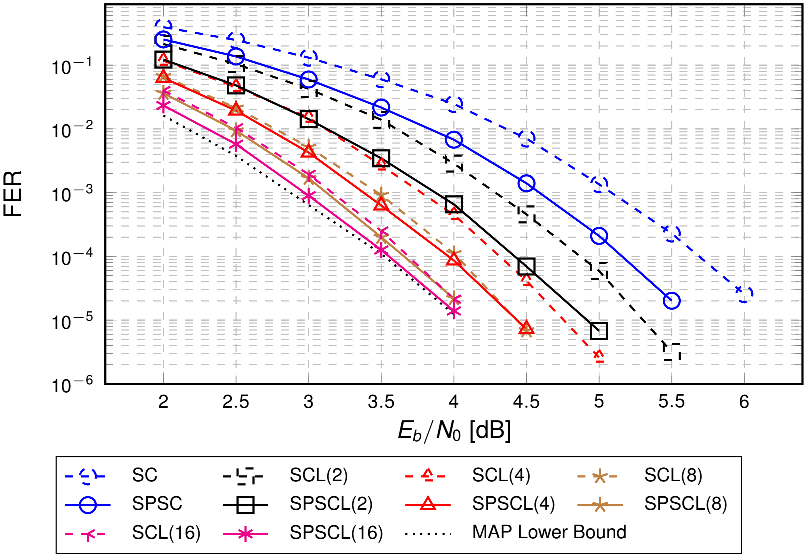
# Complexity Analysis

- Memory: Same as SC
- Parallel implementation:
  - Computational complexity: Roughly $n$ times SC
  - Latency: Same as SC
- Serial implementation:
  - Computational complexity: Same as SC
  - Latency: Roughly $n$ times SC

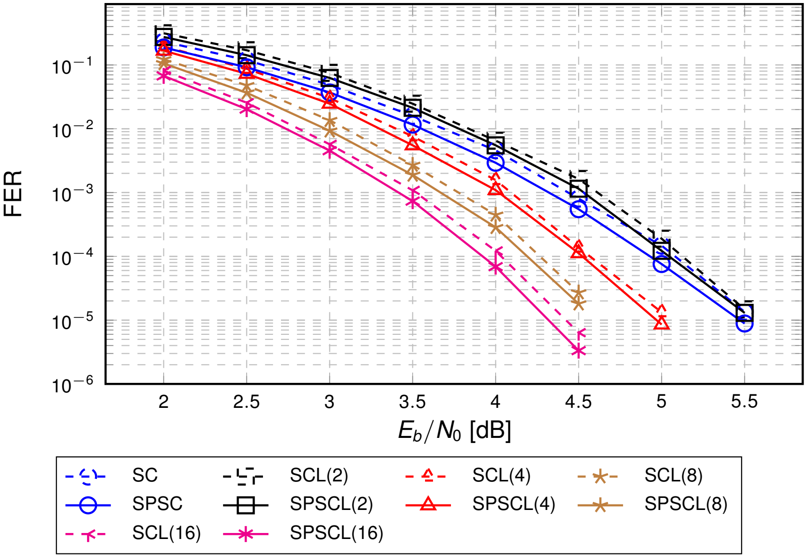
# Performance Analysis

- RM Codes:
  - Permutations do not change the info/frozen bit pattern
  - Code is unchanged but better bits are decoded earlier
  - Significant improvement is expected
- Polar Codes:
  - Permutations may result in a different info/frozen bit pattern not optimized for the channel
  - We only apply SP on RM sub-codes of polar codes
  - The info/frozen bit pattern remains unchanged

# Results: $\mathcal{RM}(128, 64)$



# Results: $\mathcal{P}(128, 64)$



# Conclusion

- We derived total number of permutations for RM and polar codes
- We proposed successive factor graph permutations for RM and polar codes to find the suitable permutations on the fly
- At FER $= 10^{-4}$:
  - SP gains up to 0.5 dB over SC and SCL of $\mathcal{RM}(128, 64)$
  - SP gains up to 0.1 dB over SC and SCL of $\mathcal{P}(128, 64)$
- SCL(16) with SP is within 0.05 dB of the MAP lower bound of $\mathcal{RM}(128, 64)$

Thank you!