Lecture 5: Polar Codes

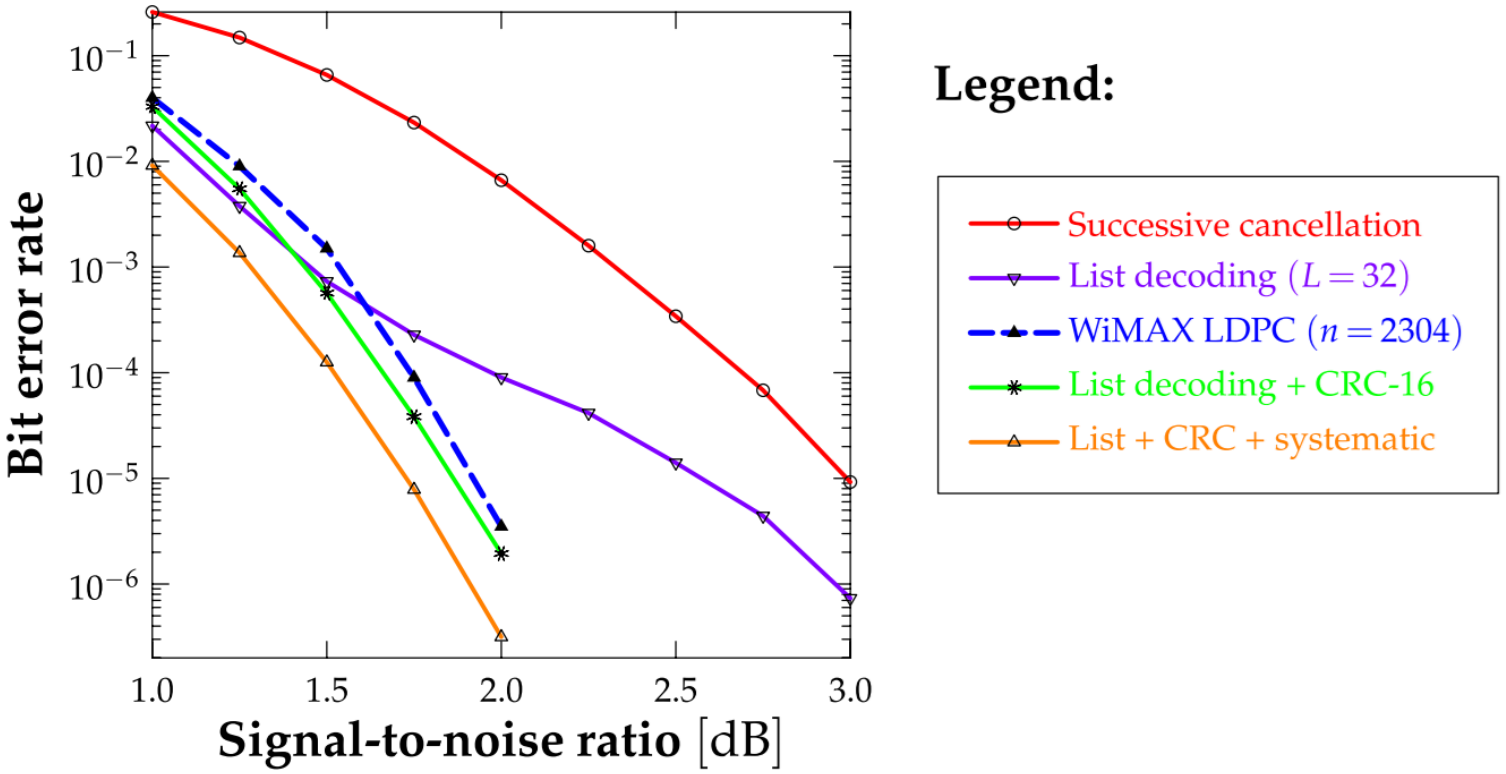
- Introduction
- Channel Polarization
  - Channel Combining
  - Channel Splitting
- Encoding
- Decoding
  - Successive Cancellation Decoding
    - List SC, CRC-Aided LSC, Simulation
  - Belief Propagation Decoding
- Summary

Introduction

- Polar codes were proposed by Erdal Arıkan.
- Capacity-achieving codes for any symmetric binary-input discrete memoryless channels (B-DMCs).
- Low encoding and decoding complexity.
- To improve their performance, polar codes are concatenated with a cyclic redundancy check (CRC) as an outer code and decoded using the list decoding algorithm (list-CRC)
- Better than turbo or LDPC codes for short polar codes.
- Polar code is considered as a candidate in 5G.

Introduction

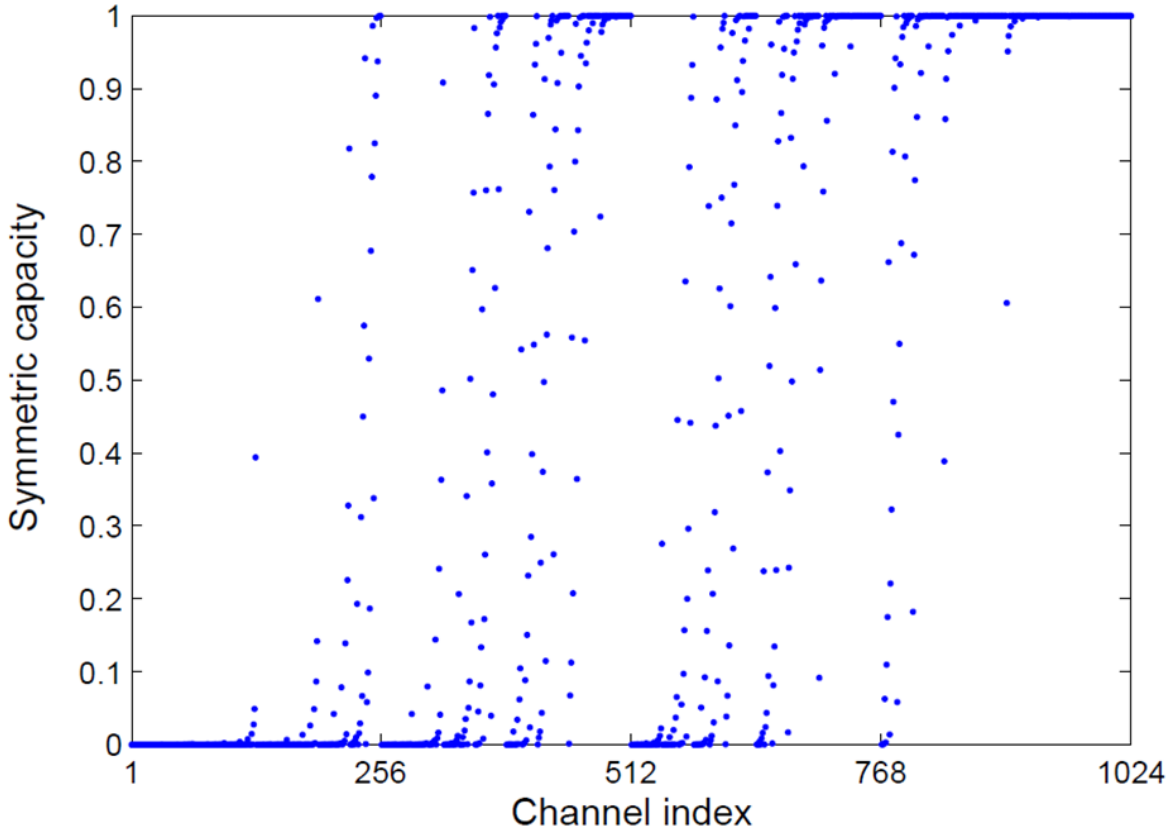
- Simulation over the BPSK-AWGN channel for polar codes of length $n = 2048$ with rate 0.5.



Channel Polarization

Channel Polarization

- $I(W_N^{(i)})$ versus $i = 1, \ldots, N = 2^{10}$ for a BEC with $\varepsilon = 0.5$.

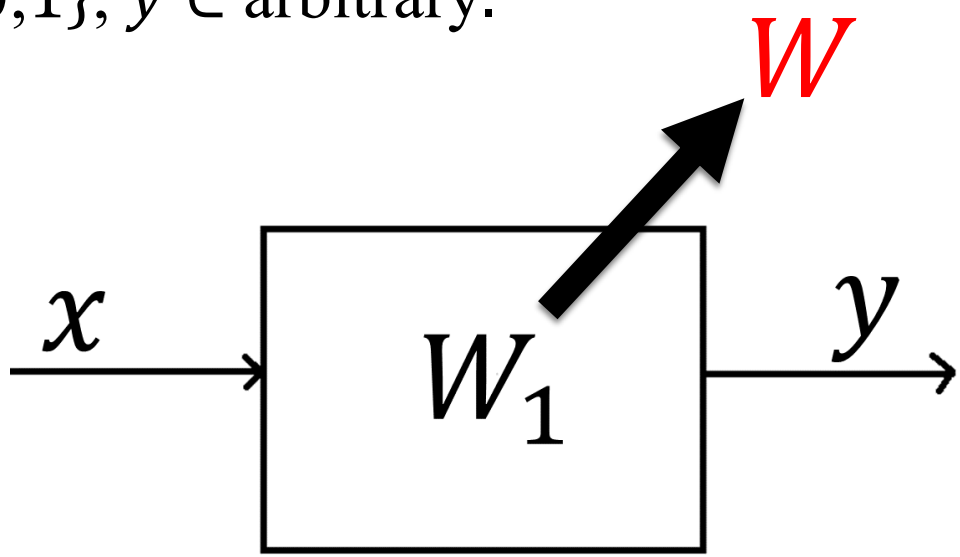


Channel Polarization

- Channel polarization is an operation by which one manufactures out of N independent copies of a given B-DMC W.
- A second set of N channels $\left\{W_N^{(i)} : 1 \leq i \leq N\right\}$ that show a polarization effect in the sense that, as N becomes large, the symmetric capacity terms $I\left(W_N^{(i)}\right)$ tend towards 0 or 1 for all but a vanishing fraction of indices i.
- This operation consists of a **channel combining phase** and a **channel splitting phase**.

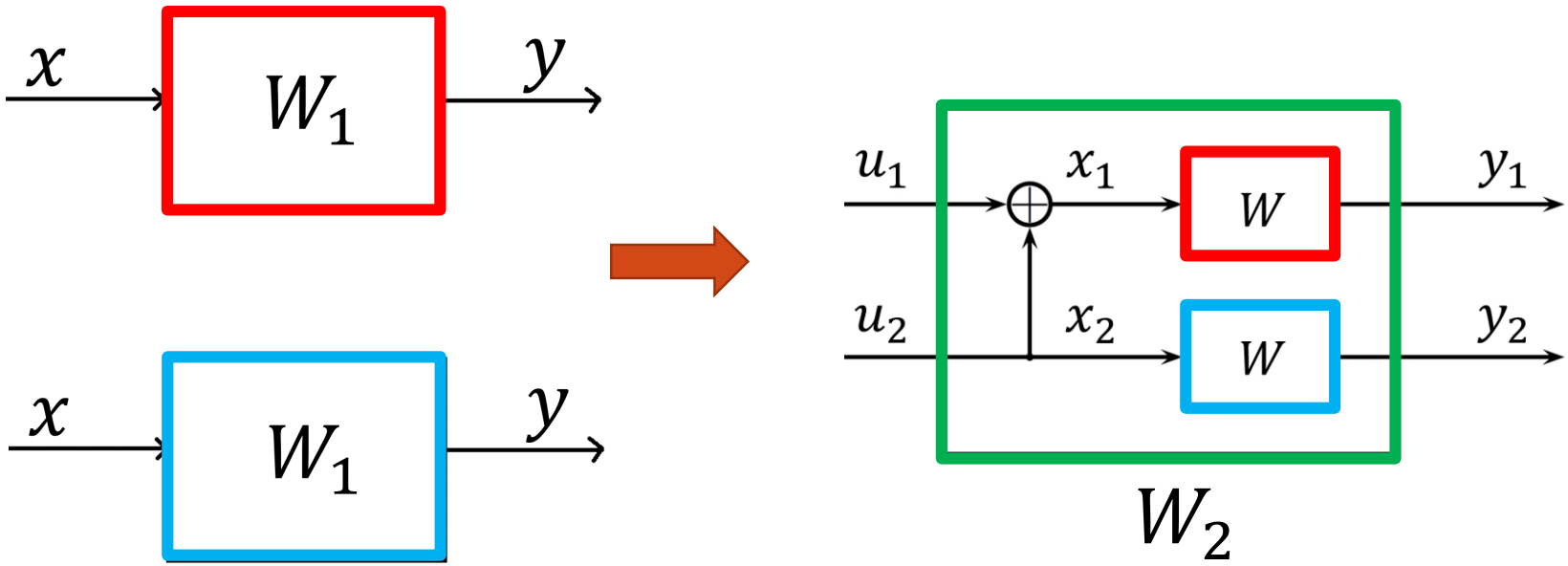
Channel Combining — W_1

- The recursion begins at the 0-th level (n = 0) with only one copy of W and we set $W_1 \triangleq W$.
- $W : X \to Y$ denotes a generic B-DMC with input alphabet X, output alphabet Y, and transition probabilities $W(y|x)$, $x \in X$, $y \in Y$.
- $x \in \{0,1\}$, $y \in$ arbitrary.



Channel Combining — $W_1 \to W_2$

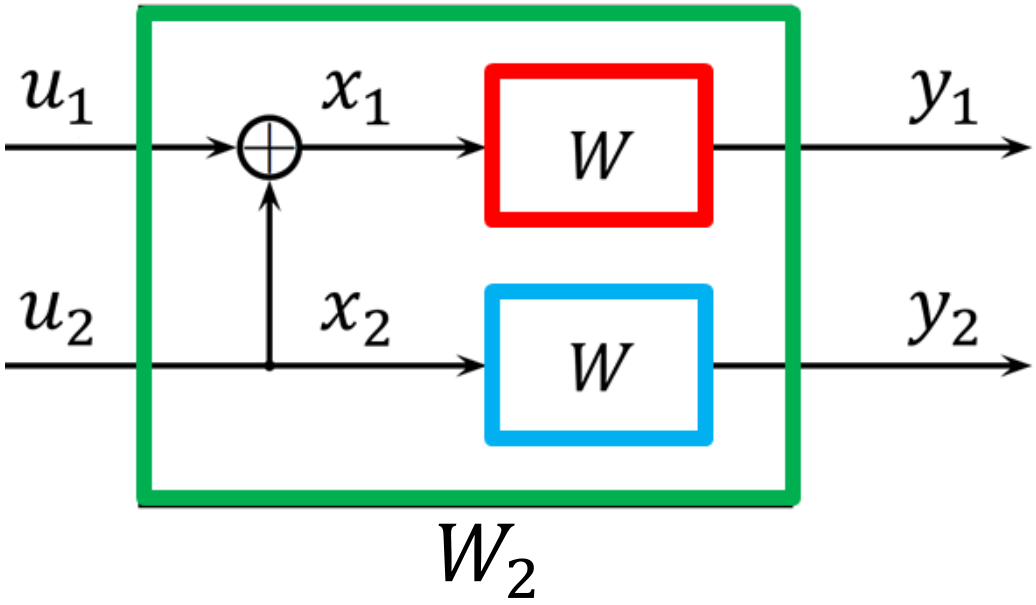
- Combine two independent copies of W_1 to obtain W_2.



Channel Combining — W_2

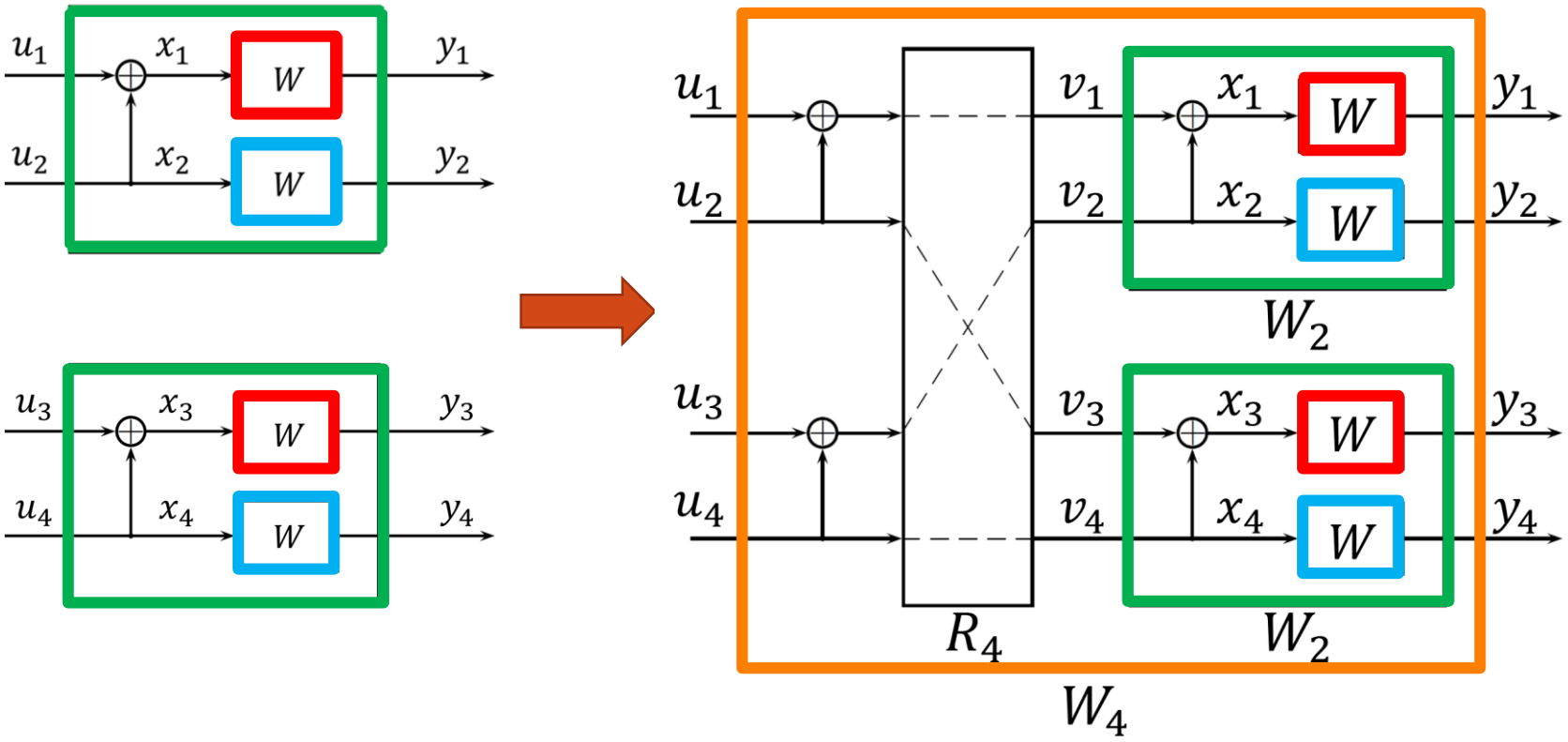
- The first level (n = 1) of the recursion combines two independent copies of W_1 and obtains the channel $W_2 : X^2 \rightarrow Y^2$ with the transition probabilities

$$W_2(y_1,y_2|u_1,u_2) = W(y_1|u_1 \oplus u_2)W(y_2|u_2)$$
$$= W(y_1|x_1)W(y_2|x_2)$$



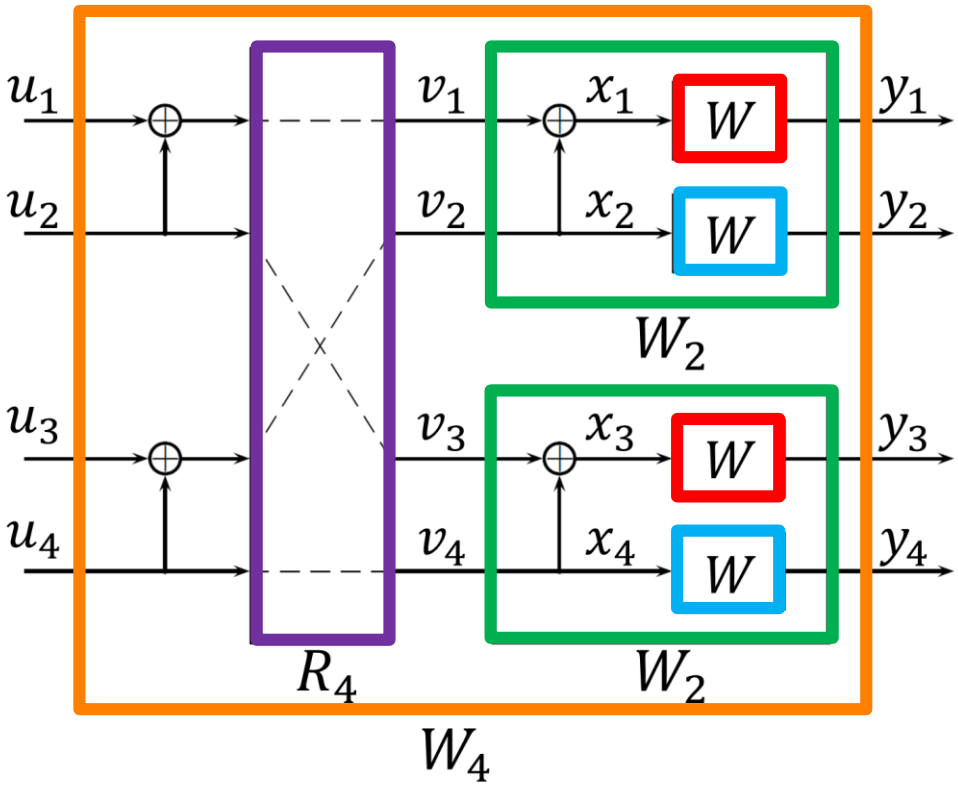
Channel Combining — $W_2 \rightarrow W_4$

- Combine two independent copies of W_2 to obtain W_4.



Channel Combining — Permutation Operation

- R_4 is a permutation operation.
- $R_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

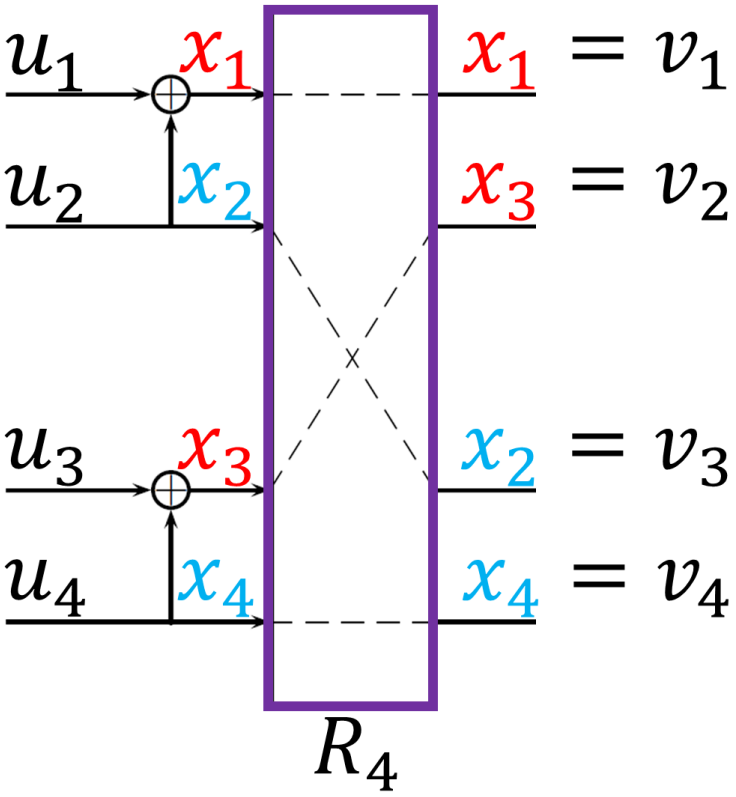


Channel Combining — Permutation Operation

- $a_1^N = (a_1, \ldots, a_N)$.
- $a_i^j = (a_i, \ldots, a_j)$, for $1 \leq i, j \leq N$, if $j < i$, a_i^j is regarded as void.
- $a_{1,o}^j$ denotes the subvector with odd indices: $(a_k : 1 \leq k \leq j; k \text{ odd})$.
- $a_{1,e}^j$ denotes the subvector with even indices: $(a_k : 1 \leq k \leq j; k \text{ even})$.
- 0_1^N denotes the all zero vector.
- Example: $a_1^5 = (5,4,6,2,1)$
  $a_2^4 = (4,6,2)$, $a_{1,e}^4 = (4,2)$, $a_{1,o}^4 = (5,6)$, $a_1^1 = (5)$

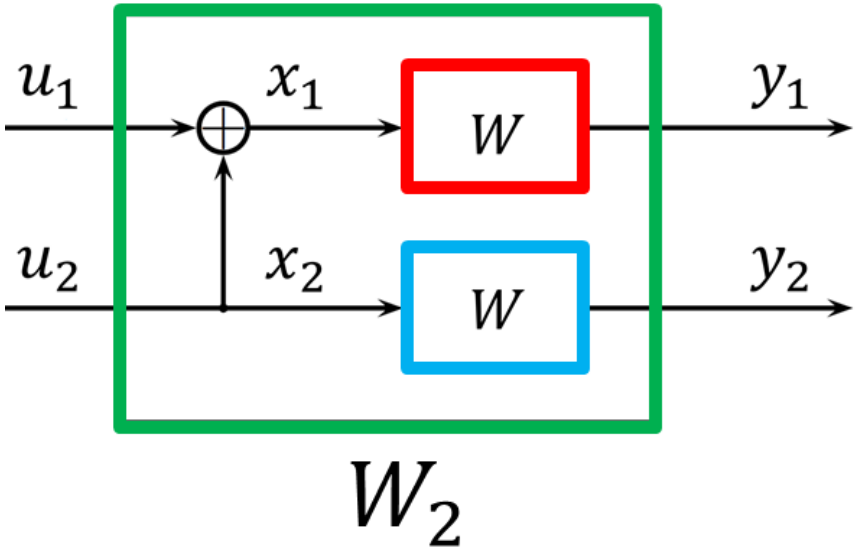
Channel Combining — Permutation Operation

- The operator R_N is a permutation, known as the reverse shuffle operation, and acts on its input x_1^N to produce $v_1^N = (x_1, x_3, \ldots, x_{N-1}, x_2, x_4, \ldots, x_N)$.



Channel Combining — Permutation Operation

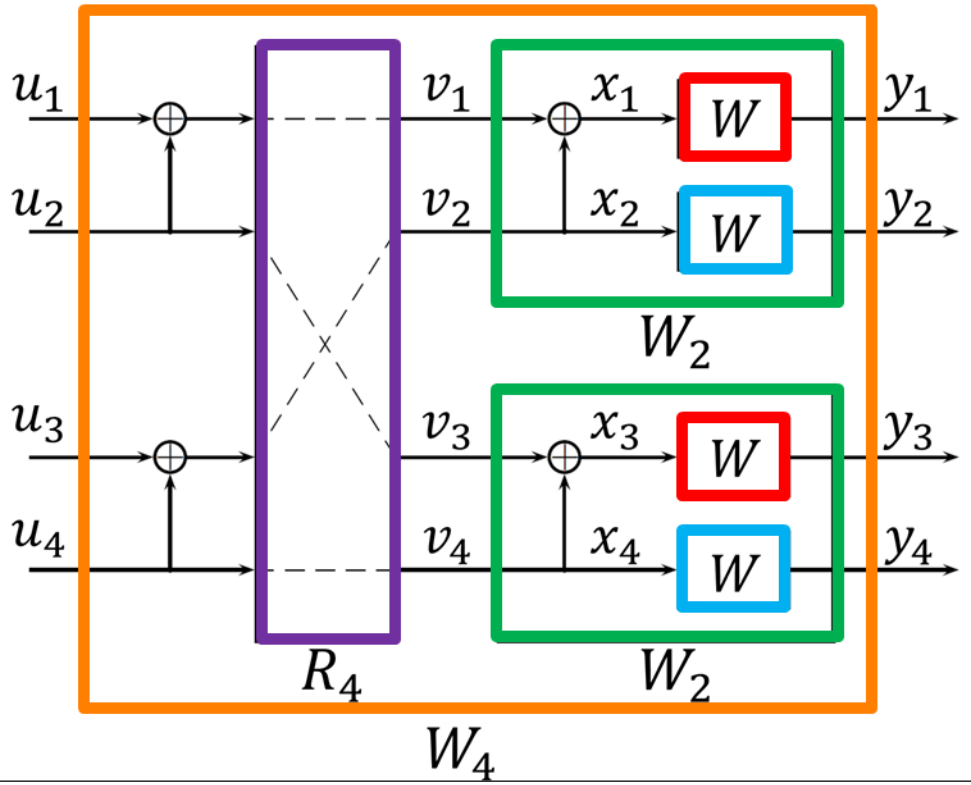
- Remind that if $N = 2$, vector (x_1, x_2) after the permutation operation is still the same as (x_1, x_2).
- Therefore, $R_2 = I_2$.



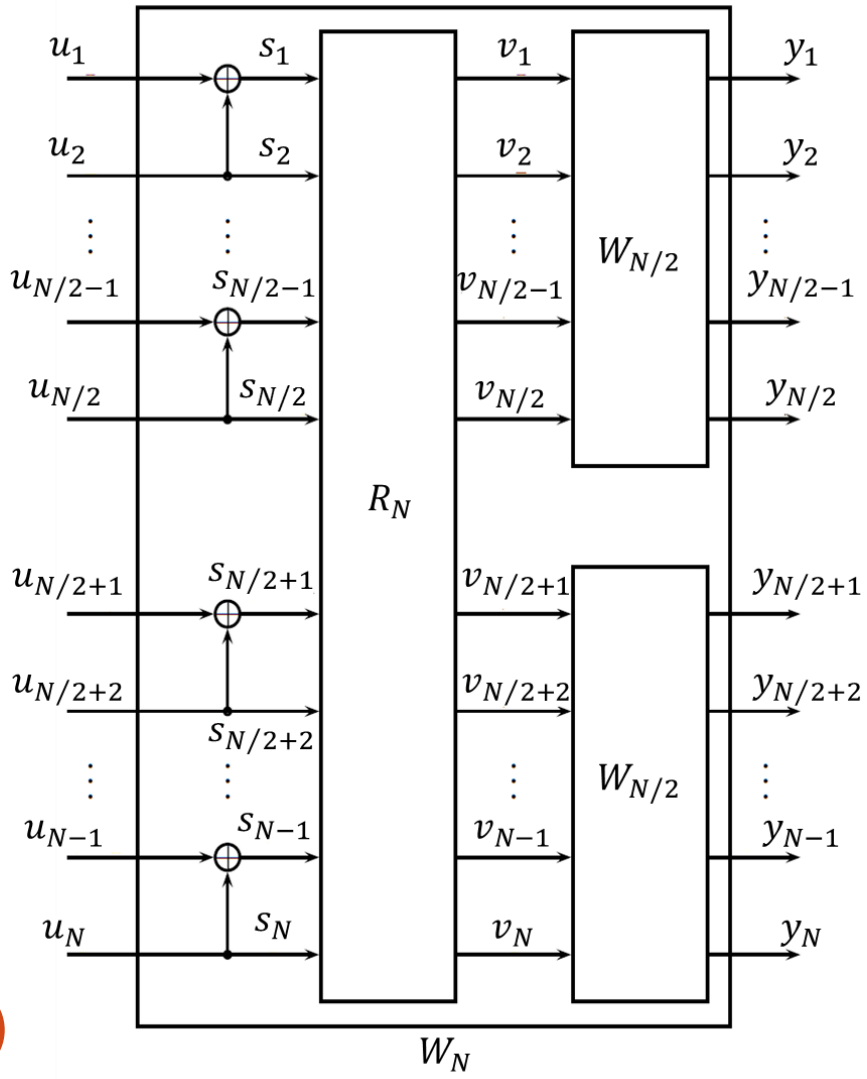
Channel Combining — W_4

- The next level of the recursion combines two independent copies of W_2 to create the channel $W_4 : X^4 \rightarrow Y^4$ with transition probabilities:

$$W_4(y_1^4 \mid u_1^4) = W_2(y_1^2 \mid u_1 \oplus u_2, u_3 \oplus u_4) W_2(y_3^4 \mid u_2, u_4)$$



Channel Combining — W_N



- $W_N : X^N \rightarrow Y^N$ with transition probabilities:

$$W_N(y_1^N \mid u_1^N)$$

$$= W_{N/2}\left(y_1^{N/2} \middle| u_{1,e}^N \oplus u_{1,o}^N\right)$$

$$\cdot W_{N/2}(y_{N/2+1}^N \mid u_{1,e}^N)$$

- N can be any power of 2

Channel Splitting — W_N

- Having synthesized the vector channel W_N out of W^N, the next step of channel polarization is to split W_N back into a set of N binary-input coordinate channels $W_N^{(i)} : X \to Y^N \times X^{i-1}$, $1 \leq i \leq N$, defined by the transition probabilities

$$W_N^{(i)}\left(y_1^N, u_1^{i-1} \middle| u_i\right) \triangleq \sum_{u_{i+1}^N \in X^{N-i}} \frac{1}{2^{N-1}} \mathrm{W}_N\left(y_1^N \mid u_1^N\right)$$

where (y_1^N, u_1^{i-1}) denotes the output of $W_N^{(i)}$ and u_i denotes its input.

Channel Splitting — W_N

- For any $n \geq 0$, $N = 2^n$, $1 \leq i \leq N$:

$$W_{2N}^{(2i-1)}\left(y_1^{2N}, u_1^{2i-2} \middle| u_{2i-1}\right)$$

$$= \sum_{u_{2i}} \frac{1}{2} W_N^{(i)}\left(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2} \middle| u_{2i-1} \oplus u_{2i}\right) W_N^{(i)}\left(y_{N+1}^{2N}, u_{1,e}^{2i-2} \middle| u_{2i}\right)$$

$$W_{2N}^{(2i)}\left(y_1^{2N}, u_1^{2i-1} \middle| u_{2i}\right)$$

$$= \frac{1}{2} W_N^{(i)}\left(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2} \middle| u_{2i-1} \oplus u_{2i}\right) W_N^{(i)}\left(y_{N+1}^{2N}, u_{1,e}^{2i-2} \middle| u_{2i}\right)$$

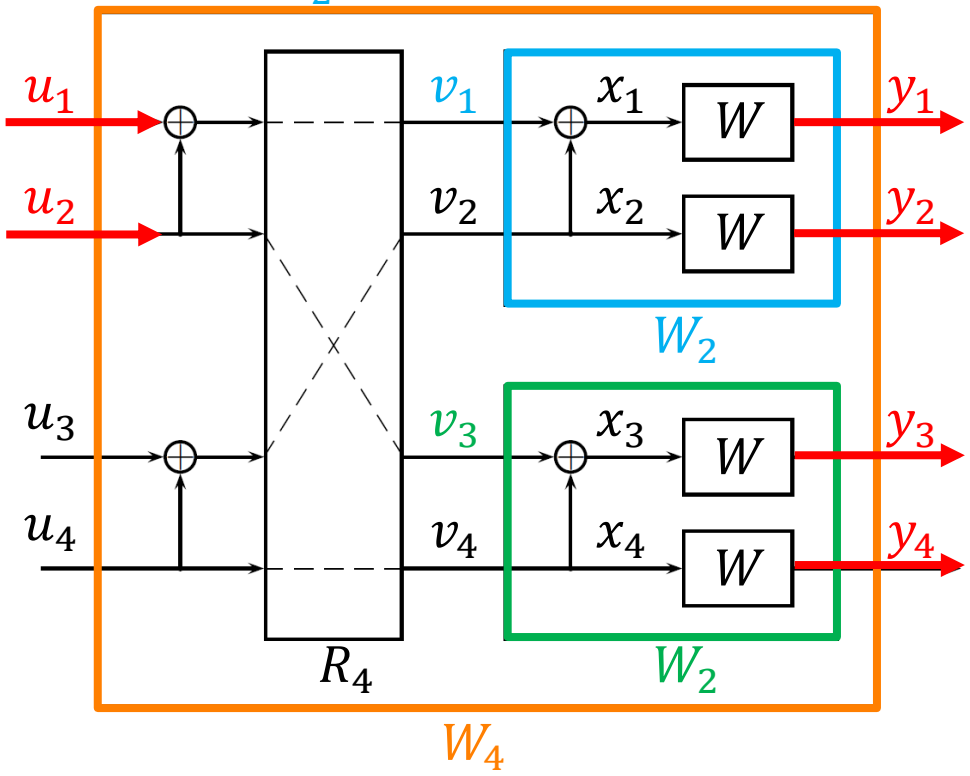
where (y_1^{2N}, u_1^{2i-2}) and (y_1^{2N}, u_1^{2i-1}) denote the output of $W_{2N}^{(2i-1)}$ and $W_{2N}^{(2i)}$, u_{2i-1} and u_{2i} denote its input, respectively.

Channel Splitting — $W_4^{(i)} \rightarrow W_2^{(i)}$

- For $N = 2, i = 1$:

$$W_4^{(1)}(y_1^4 | u_1) = \sum_{u_2} \frac{1}{2} W_2^{(1)}(y_1^2 | u_1 \oplus u_2)\, W_2^{(1)}(y_3^4 | u_2)$$

$$W_4^{(2)}(y_1^4, u_1 | u_2) = \frac{1}{2} W_2^{(1)}(y_1^2 | u_1 \oplus u_2)\, W_2^{(1)}(y_3^4 | u_2)$$

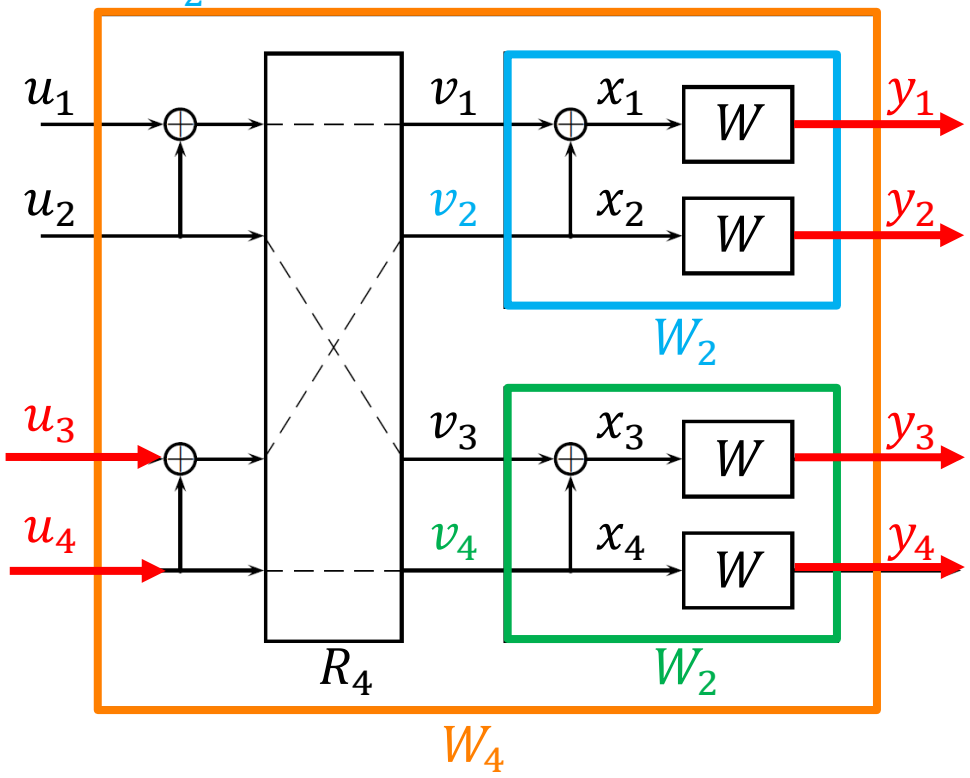


Channel Splitting — $W_4^{(i)} \rightarrow W_2^{(i)}$

- For $N = 2, i = 2$:

$$W_4^{(3)}(y_1^4, u_1^2 | u_3) = \sum_{u_4} \frac{1}{2} W_2^{(2)}(y_1^2, u_1 \oplus u_2 | u_3 \oplus u_4)\, W_2^{(2)}(y_3^4, u_2 | u_4)$$

$$W_4^{(4)}(y_1^4, u_1^3 | u_4) = \frac{1}{2} W_2^{(2)}(y_1^2, u_1 \oplus u_2 | u_3 \oplus u_4)\, W_2^{(2)}(y_3^4, u_2 | u_4)$$

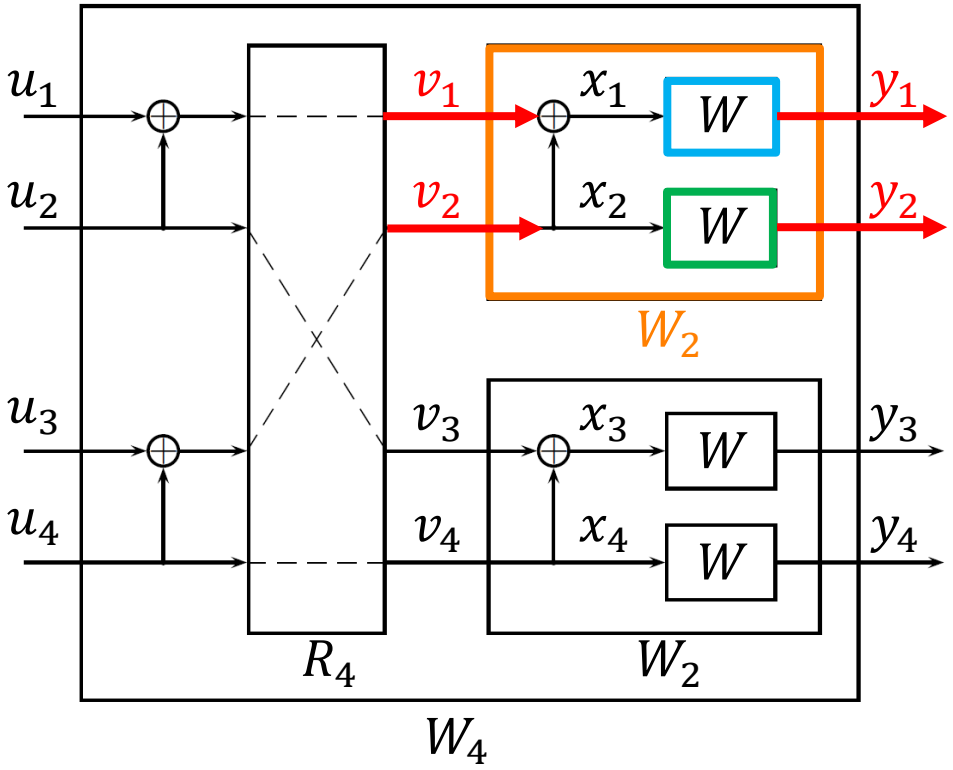


Channel Splitting — $W_2^{(i)} \rightarrow W_1^{(1)}$

- For $N = 1, i = 1$:

$$W_2^{(1)}(y_1^2 | v_1) = \sum_{v_2} \frac{1}{2} W_1^{(1)}(y_1 | v_1 \oplus v_2)\, W_1^{(1)}(y_2 | v_2)$$

$$W_2^{(2)}(y_1^2, v_1 | v_2) = \frac{1}{2} W_1^{(1)}(y_1 | v_1 \oplus v_2)\, W_1^{(1)}(y_2 | v_2)$$



Channel Polarization

- Given a B-DMC W, there are two channel parameters:
  - Symmetric capacity:

$$I(W) \triangleq \sum_{y \in Y} \sum_{x \in X} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)}$$

  - Bhattacharyya parameter:

$$Z(W) \triangleq \sum_{y \in Y} \sqrt{W(y|0)W(y|1)}$$

- These parameters are used as measures of rate and reliability, respectively.

Channel Polarization

- $I(W)$ is the highest rate at which reliable communication is possible across W.
- $Z(W)$ is an upper bound on the probability of ML decision error when W is used only once to transmit a 0 or 1.
- For any B-DMC W,

$$I(W) \geq \log_2 \frac{2}{1 + Z(W)}$$

$$I(W) \leq \sqrt{1 - Z(W)^2}$$

- $I(W) \approx 1$ iff $\mathrm{Z}(W) \approx 0$, and $I(W) \approx 0$ iff $\mathrm{Z}(W) \approx 1$.

Channel Polarization

- W is a BEC with erasure probability ε. The numbers $\left\{I\left(W_N^{(i)}\right)\right\}$ can be computed using the recursive relations

$$I\left(W_N^{(2i-1)}\right) = I\left(W_{N/2}^{(i)}\right)^2$$

$$I\left(W_N^{(2i)}\right) = 2I\left(W_{N/2}^{(i)}\right) - I\left(W_{N/2}^{(i)}\right)^2$$

  with $I\left(W_1^{(1)}\right) = 1 - \varepsilon$.

- This recursion is valid only for BECs. No efficient algorithm is known for calculation of $\{I\left(W_N^{(i)}\right)\}$ for a general B-DMC W.

Channel Polarization

- The parameters $\left\{Z\left(W_N^{(i)}\right)\right\}$ can be computed by

$$Z\left(W_N^{(2i-1)}\right) = 2Z\left(W_{N/2}^{(i)}\right) - Z\left(W_{N/2}^{(i)}\right)^2$$

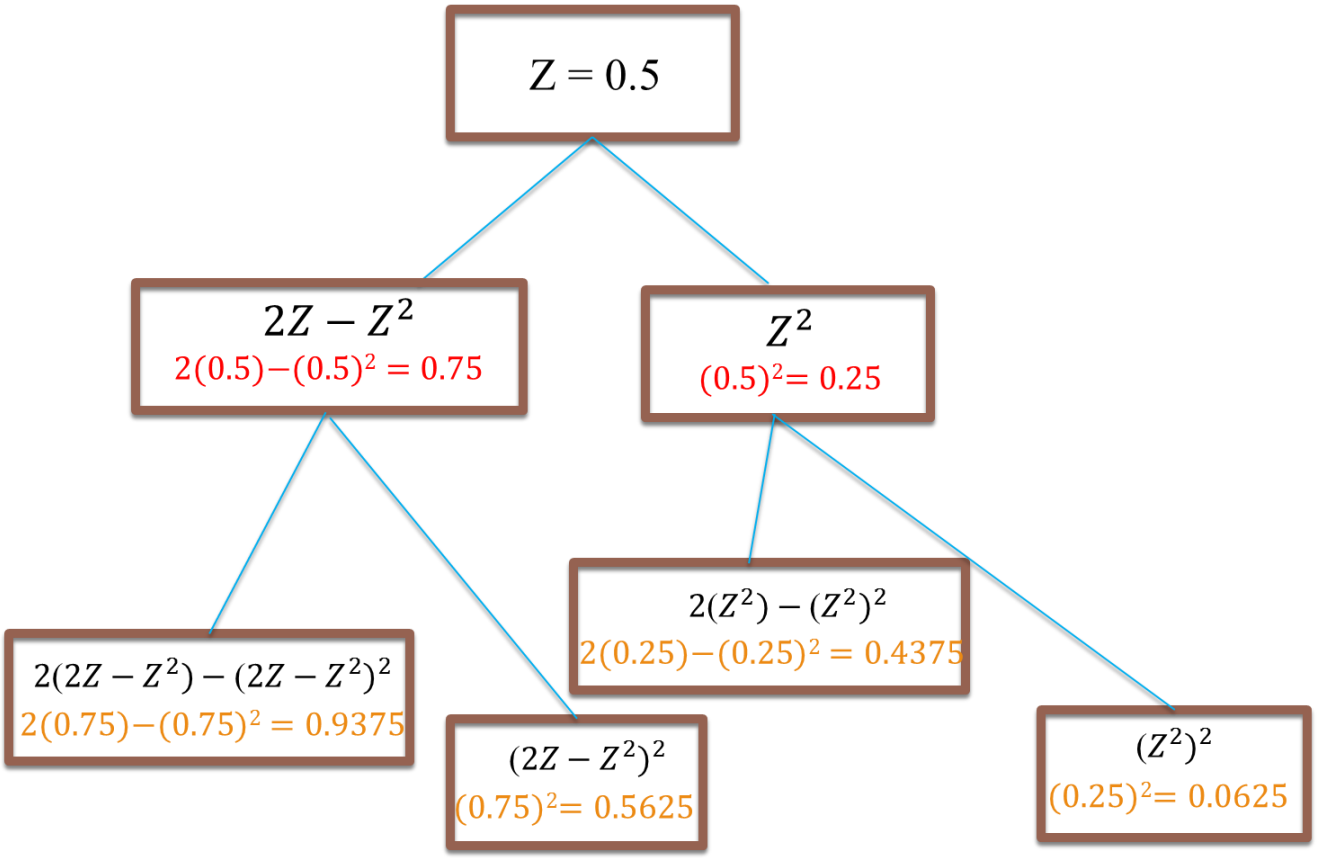
$$Z\left(W_N^{(2i)}\right) = Z\left(W_{N/2}^{(i)}\right)^2$$

  with $Z\left(W_1^{(1)}\right) = \varepsilon$.

- This recursion is valid only for BECs. The recursive relations use the fact that $I\left(W_N^{(i)}\right) = 1 - Z\left(W_N^{(i)}\right)$ for a BEC W.

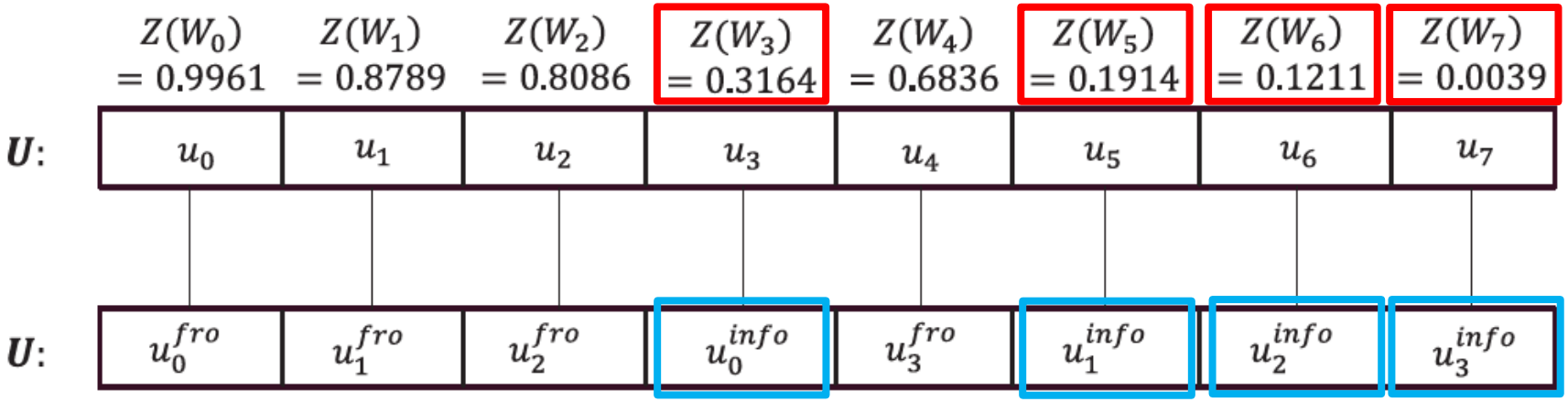
Channel Polarization

- $Z(W_N^{(i)})$
- For $N = 4, \varepsilon = 0.5$



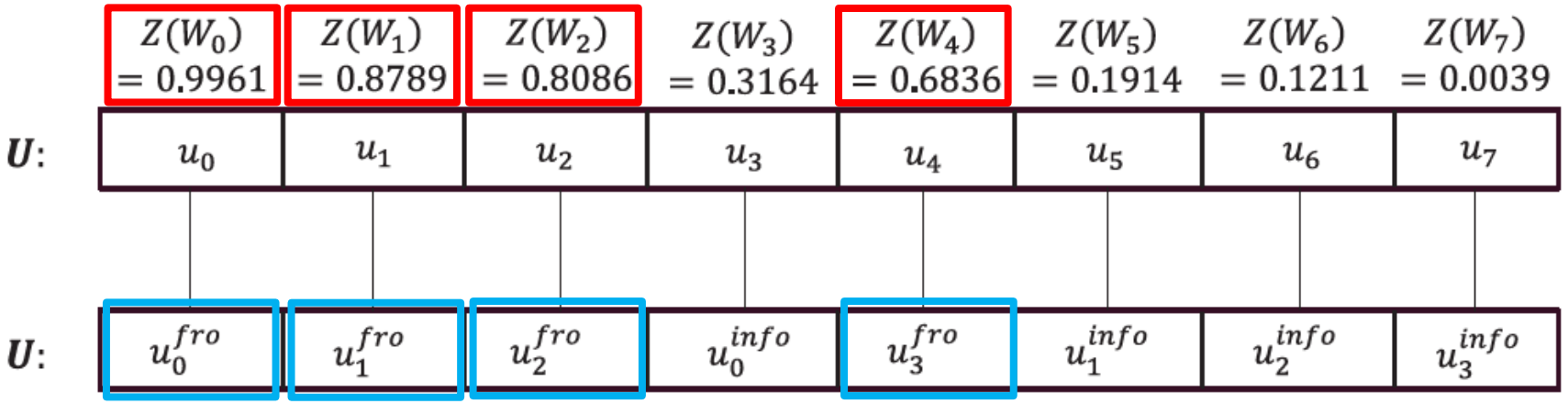
Channel Polarization

- Information set $\mathcal{A}$ is chosen as a K-element subset of $\{1,\ldots,N\}$ such that the Bhattacharyya parameters satisfy $Z\left(W_N^{(i)}\right) \leq Z\left(W_N^{(j)}\right)$ for all $i \in \mathcal{A}$ and $j \in \mathcal{A}^c$.
- For $N = 8$, code rate $= 0.5$, $K = 4$. Choose 4 smallest $Z(W)$'s to determine the information set $\mathcal{A}$.



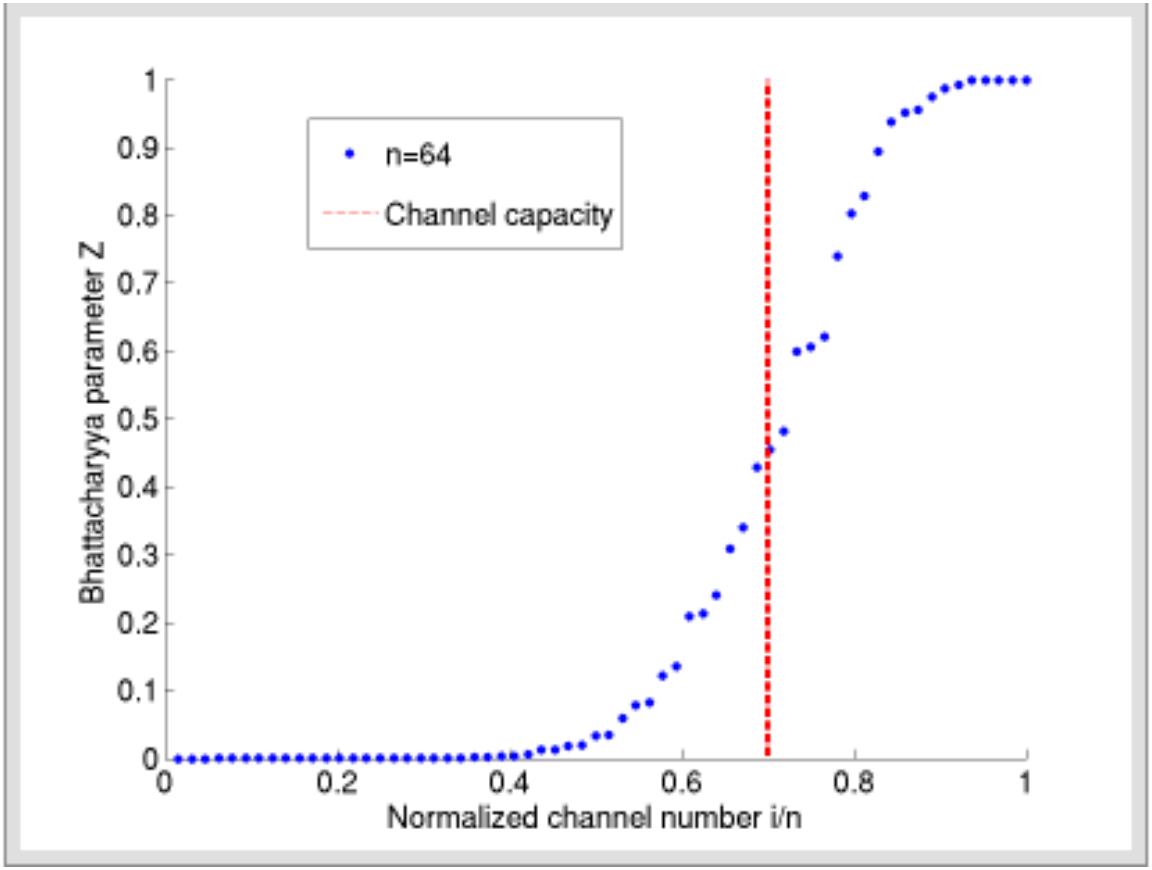
Channel Polarization

- Frozen set $\mathcal{A}^c$ is chosen as a $(N-K)$-element subset of $\{1,\ldots,N\}$, and these frozen bits are set to be "0".
- For $N=8$, code rate $=0.5$, $K=4$. Choose 4 biggest $Z(W)$'s to determine the frozen set $\mathcal{A}$.



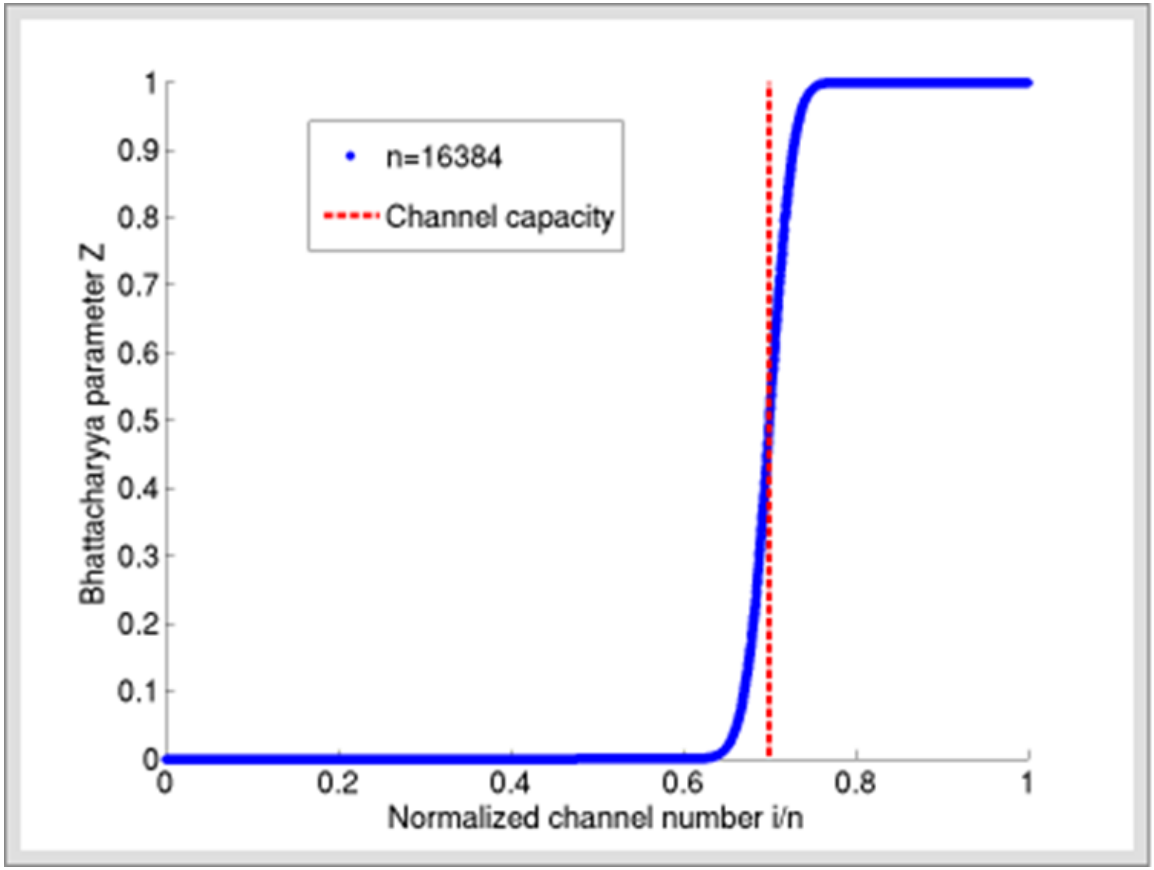
Channel Polarization

- Permuted $Z\left(W_N^{(i)}\right)$ for a BEC with $N = 64$, $\varepsilon = 0.3$.



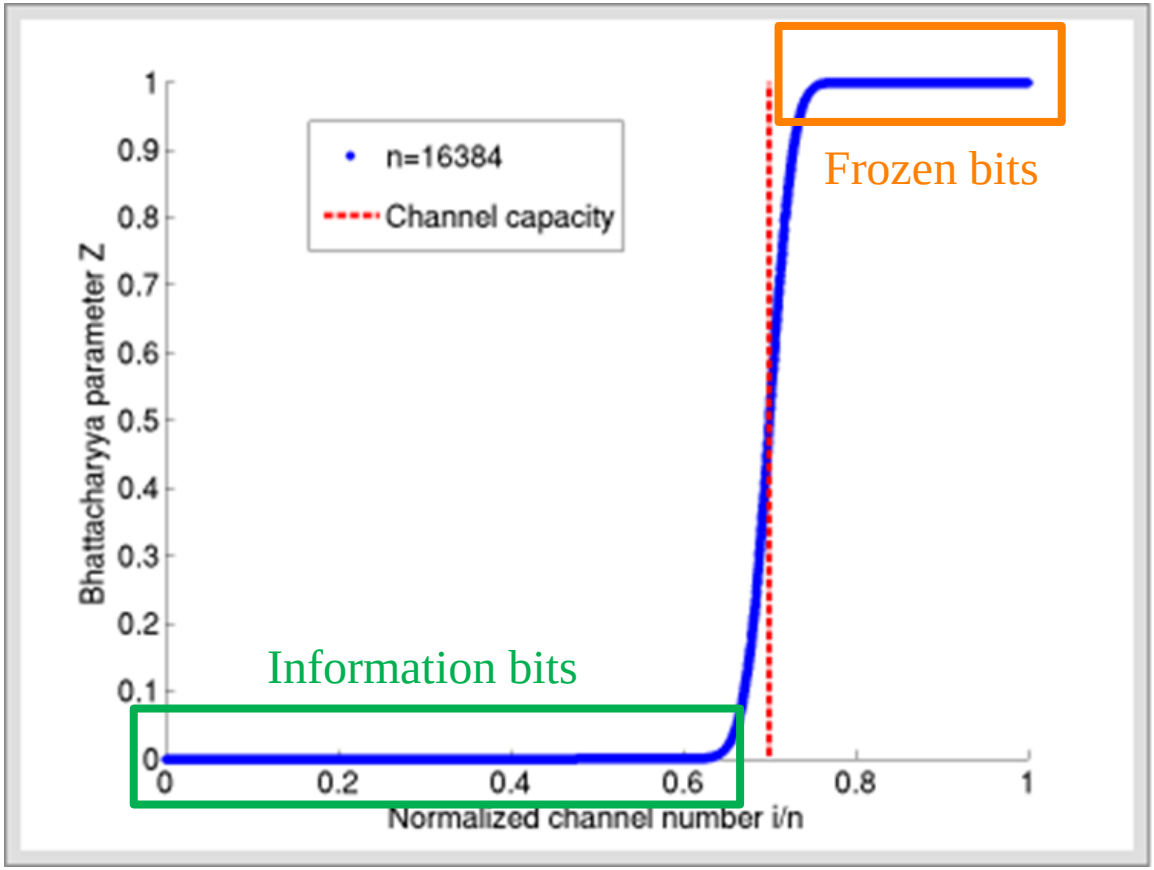
Channel Polarization

- Permuted $Z\left(W_N^{(i)}\right)$ for a BEC with $N = 16384$, $\varepsilon = 0.3$.



Channel Polarization

- Permuted $Z\left(W_N^{(i)}\right)$ for a BEC with $N = 16384$, $\varepsilon = 0.3$.



Encoding

Kronecker Product

- The Kronecker product of an m-by-n matrix $A = [A_{ij}]$ and an r-by-s matrix $B = [B_{ij}]$ is defined as

$$\mathrm{A} \otimes \mathrm{B} = \begin{bmatrix} A_{11}B & \cdots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \cdots & A_{mn}B \end{bmatrix}$$

  which is a mr-by-ns matrix.

- $A^{\otimes n} = A \otimes A^{\otimes(n-1)}$ for all $n \geq 1$.
- $A \otimes 0 = [1]$.

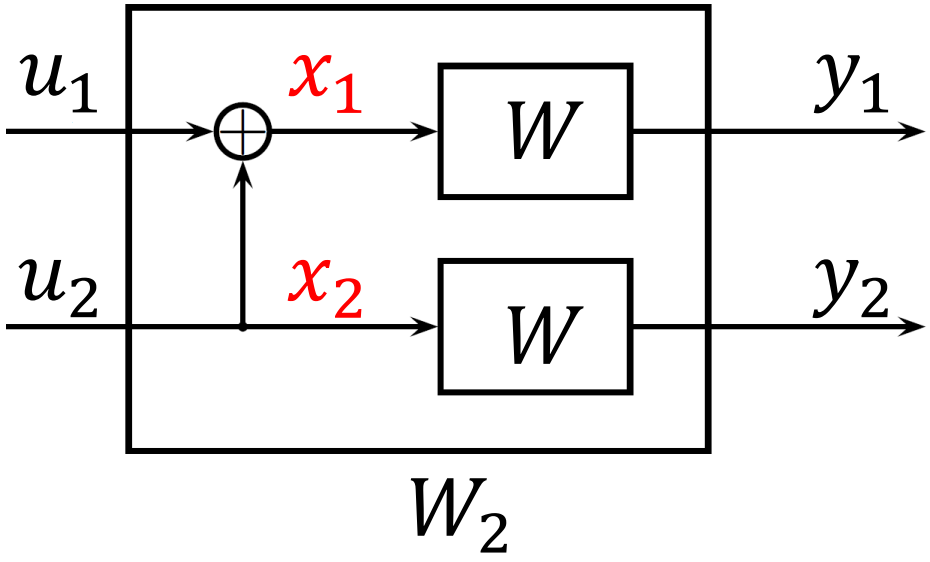
Kronecker Product

- $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.
- $F^{\otimes 2} = \begin{bmatrix} F & 0 \\ F & F \end{bmatrix}$.
- $F^{\otimes 3} = \begin{bmatrix} F^{\otimes 2} & 0 \\ F^{\otimes 2} & F^{\otimes 2} \end{bmatrix}$.
- $I_2 \otimes F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$

Construction of Polar Codes

- G_2

$$[u_1 \quad u_2]\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} = [u_1 + u_2 \quad u_2]\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$
$$= [x_1 \quad x_2].$$



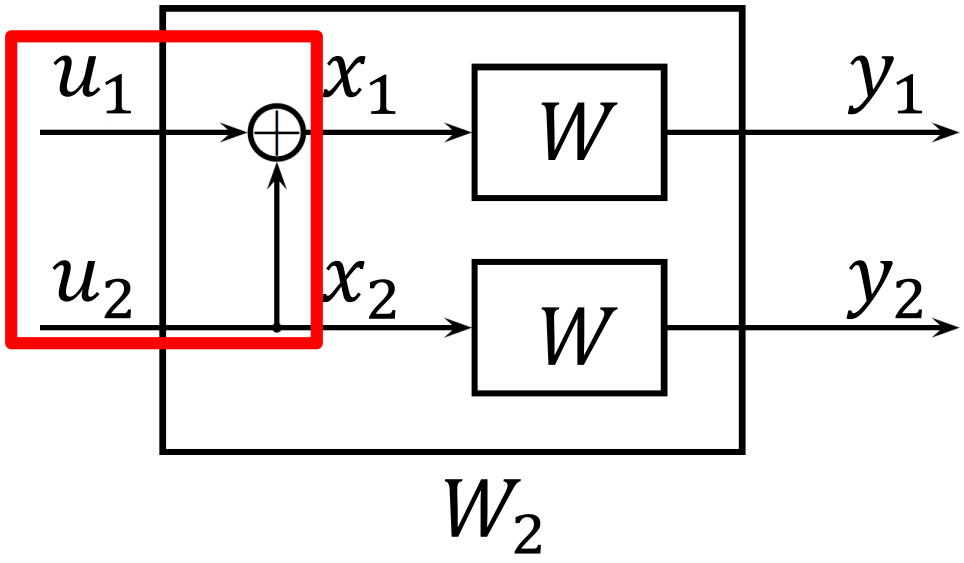
Construction of Polar Codes

- G_2

$$[u_1 \quad u_2]\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} = [u_1 + u_2 \quad u_2]\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$
$$= [x_1 \quad x_2].$$

$$\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix} = F.$$

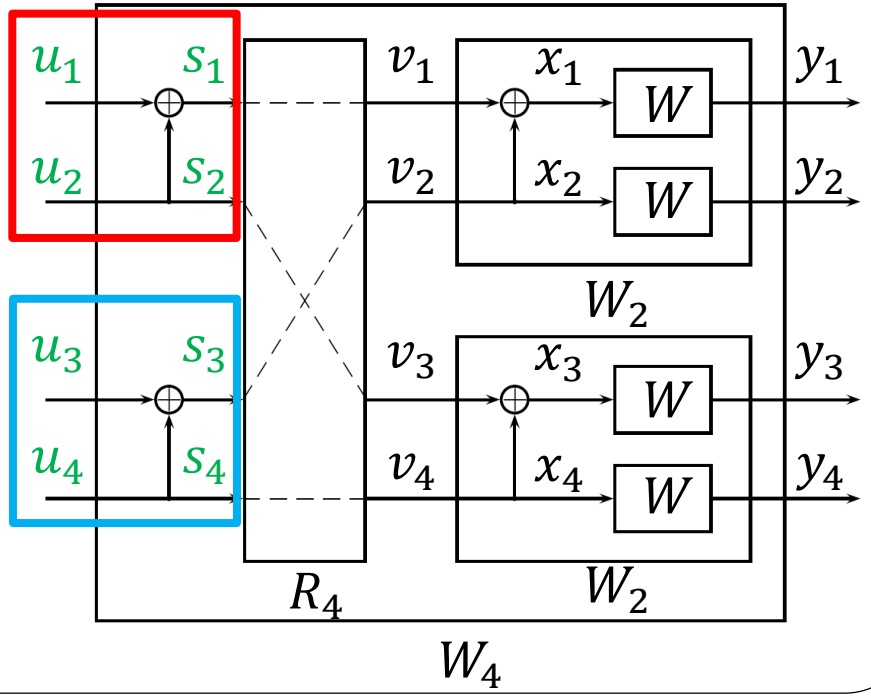
$$\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} = G_2.$$



Construction of Polar Codes

- G_4

$$[u_1 \quad u_2 \quad u_3 \quad u_4]\begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 1 & 1\end{bmatrix} = [s_1 \quad s_2 \quad s_3 \quad s_4].$$

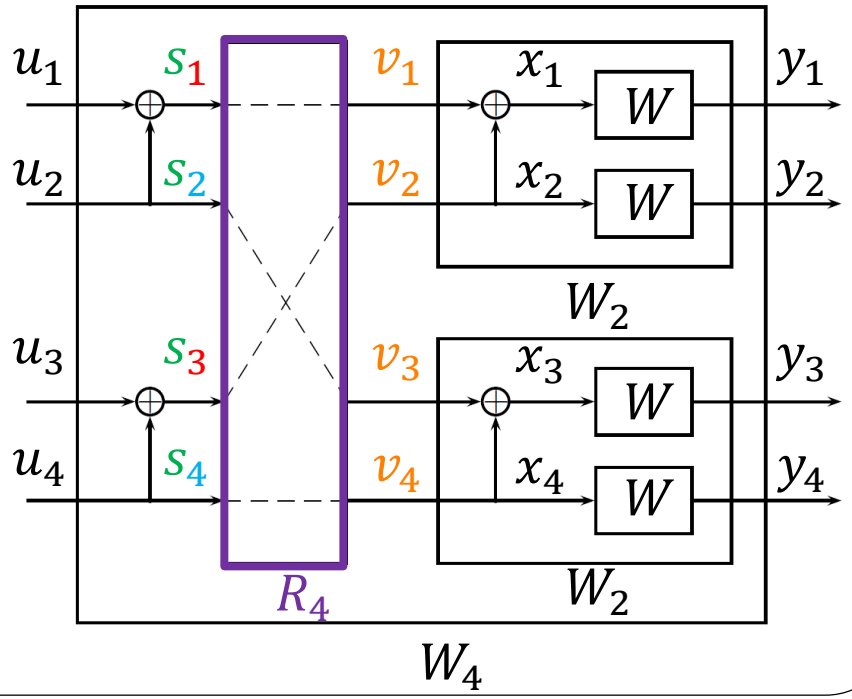


Construction of Polar Codes

- G_4

$$[s_1 \quad s_2 \quad s_3 \quad s_4]\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} = [s_1 \quad s_3 \quad s_2 \quad s_4]$$

$$= [v_1 \quad v_2 \quad v_3 \quad v_4].$$



Construction of Polar Codes

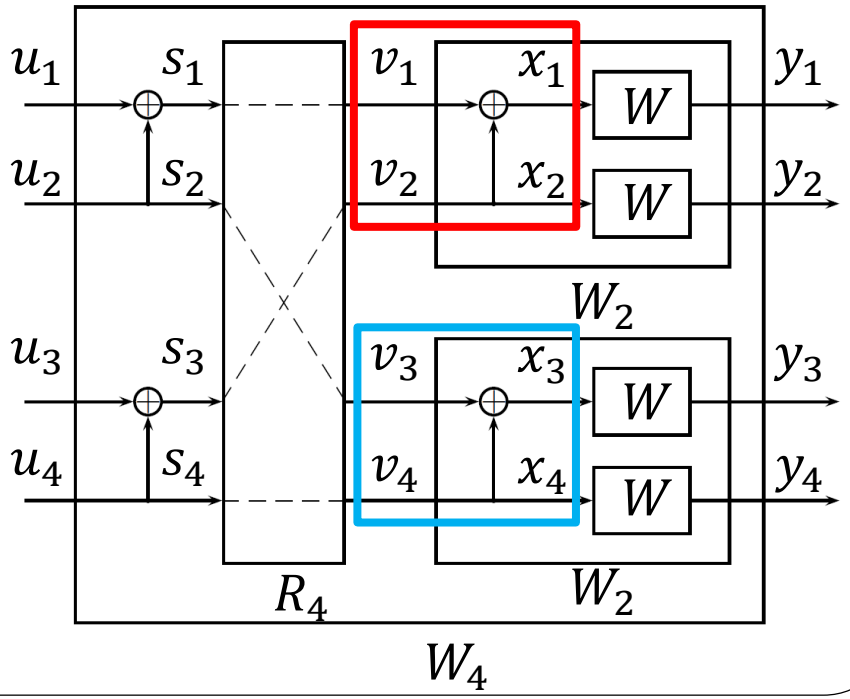
- G_4

$$[v_1 \quad v_2 \quad v_3 \quad v_4]\begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 1 & 1\end{bmatrix} = [x_1 \quad x_2 \quad x_3 \quad x_4].$$

$$\begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 1 & 1\end{bmatrix} = \begin{bmatrix}G_2 & 0\\ 0 & G_2\end{bmatrix}$$

$$= I_2 \otimes G_2.$$

$$G_2 = F = \begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}.$$



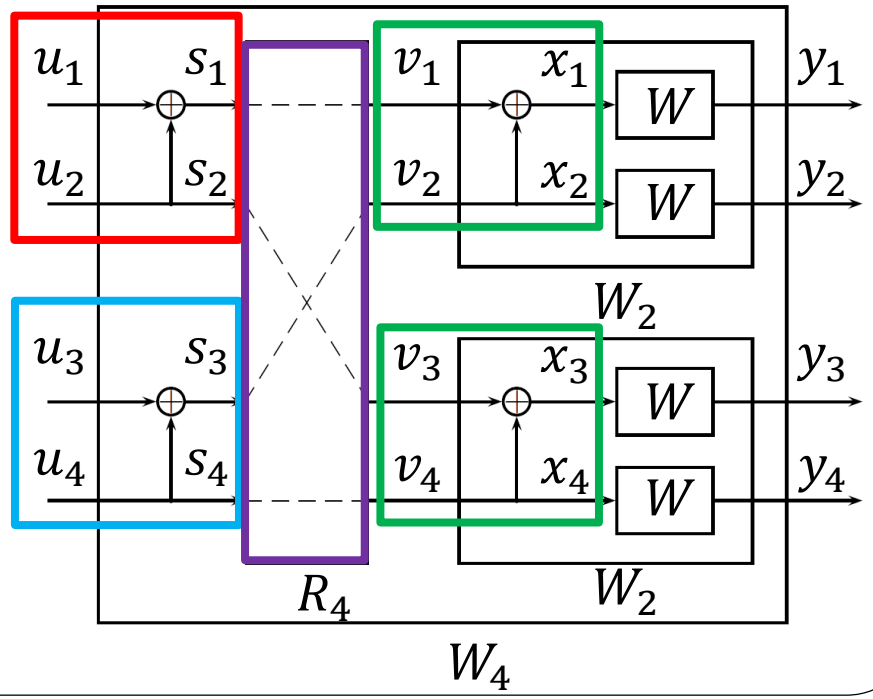
Construction of Polar Codes

$$[u_1 \quad u_2 \quad u_3 \quad u_4]\begin{bmatrix}1&0&0&0\\1&1&0&0\\0&0&1&0\\0&0&1&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}G_2&0\\0&G_2\end{bmatrix}$$

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

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

$$= G_4.$$

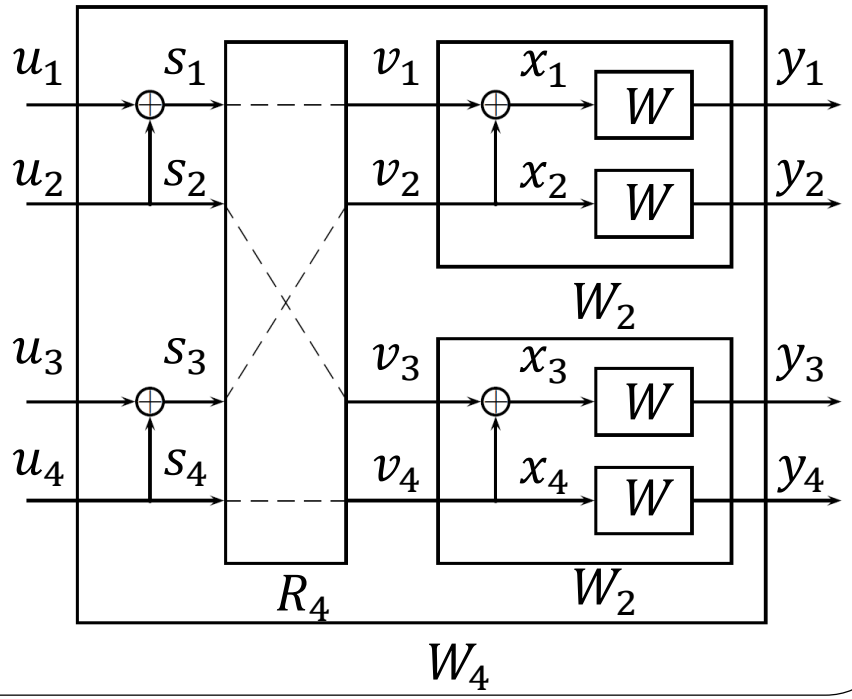


Construction of Polar Codes

$$[u_1 \quad u_2 \quad u_3 \quad u_4]\begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 0 & 1 & 0\\ 1 & 1 & 0 & 0\\ 1 & 1 & 1 & 1\end{bmatrix} = [x_1 \quad x_2 \quad x_3 \quad x_4]$$

$$\Longrightarrow u_1^4 G_4 = x_1^4.$$

where G_4 is the generator matrix.



Polar Coding

- The basic idea of polar coding is to create a coding system where one can access each coordinate channel $W_N^{(i)}$ individually and send data only through those for which $Z\left(W_N^{(i)}\right)$ is near 0.
- A block length N code is encoded as:

$$x_1^N = u_1^N G_N$$

where $N = 2^n$ for some $n \geq 0$.

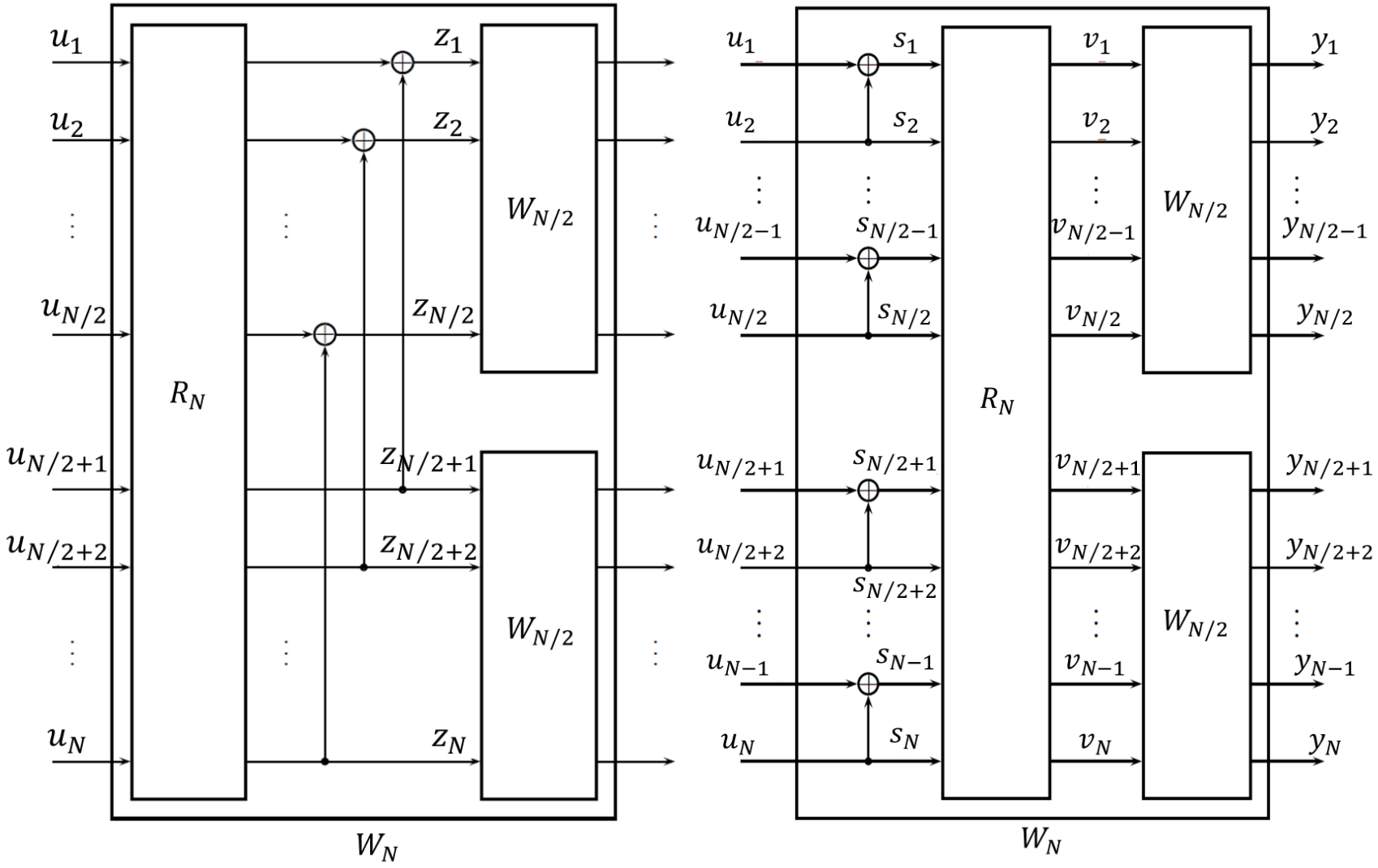
G_N-Coset Codes

- A G_N-coset code is identified by a parameter vector $(N,K,\mathcal{A},u_{\mathcal{A}^c})$.
- Set $\mathcal{A}$ as the information set and $u_{\mathcal{A}^c} \in X^{N-K}$ as frozen bits or vector.
- For a $\left(4,2,\{2,4\},(1,0)\right)$ polar code, the mapping of the encoder is

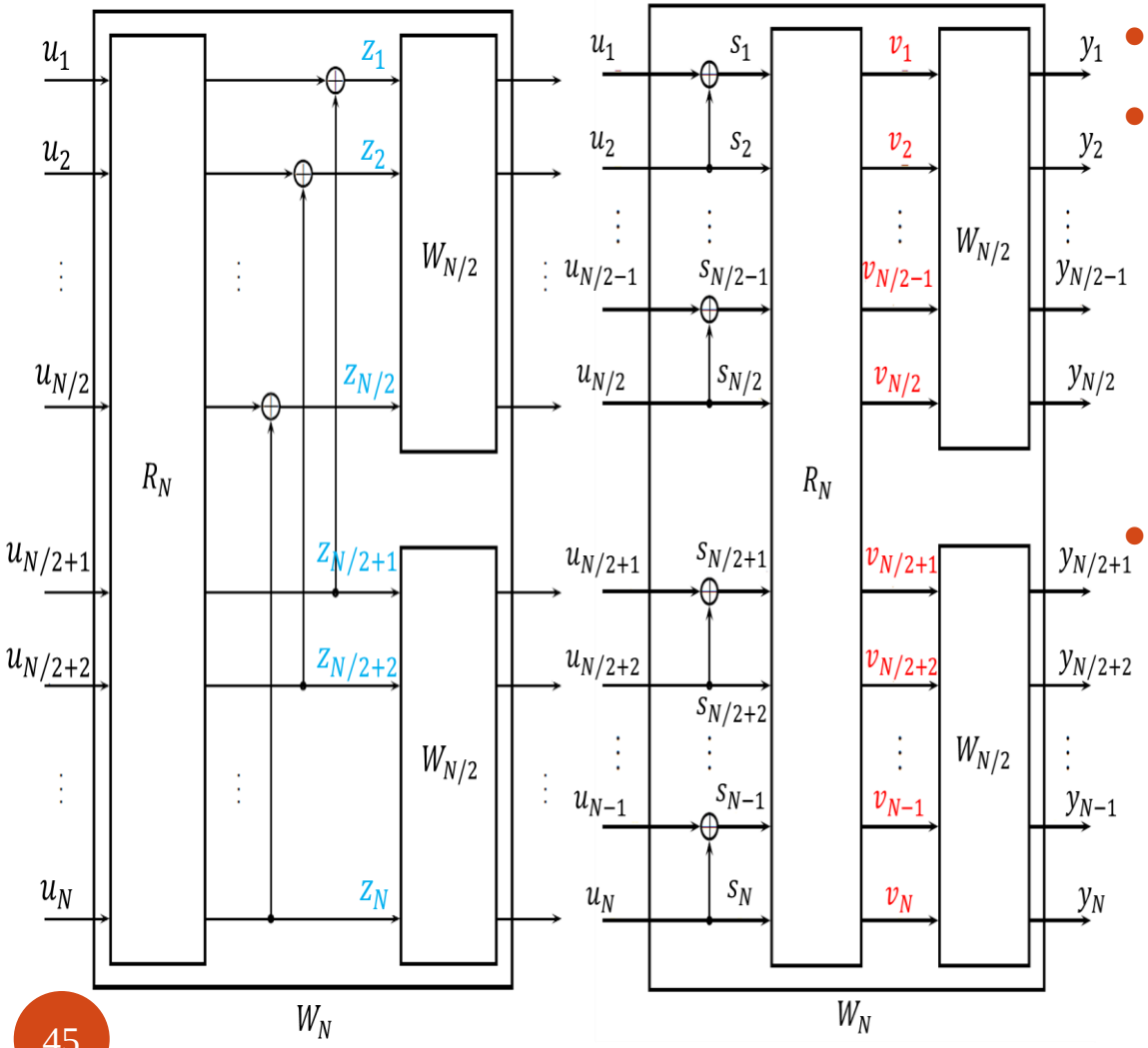
$$x_1^4 = u_1^4 G_4$$
$$= (u_2,u_4)\begin{bmatrix}1 & 0 & 1 & 0\\ 1 & 1 & 1 & 1\end{bmatrix} + (1,0)\begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\end{bmatrix}$$

For a source block $(u_2,u_4) = (1,1)$, the coded block is $x_1^4 = (1,1,0,1)$.

Alternative Realization for W_N



Alternative Realization for W_N



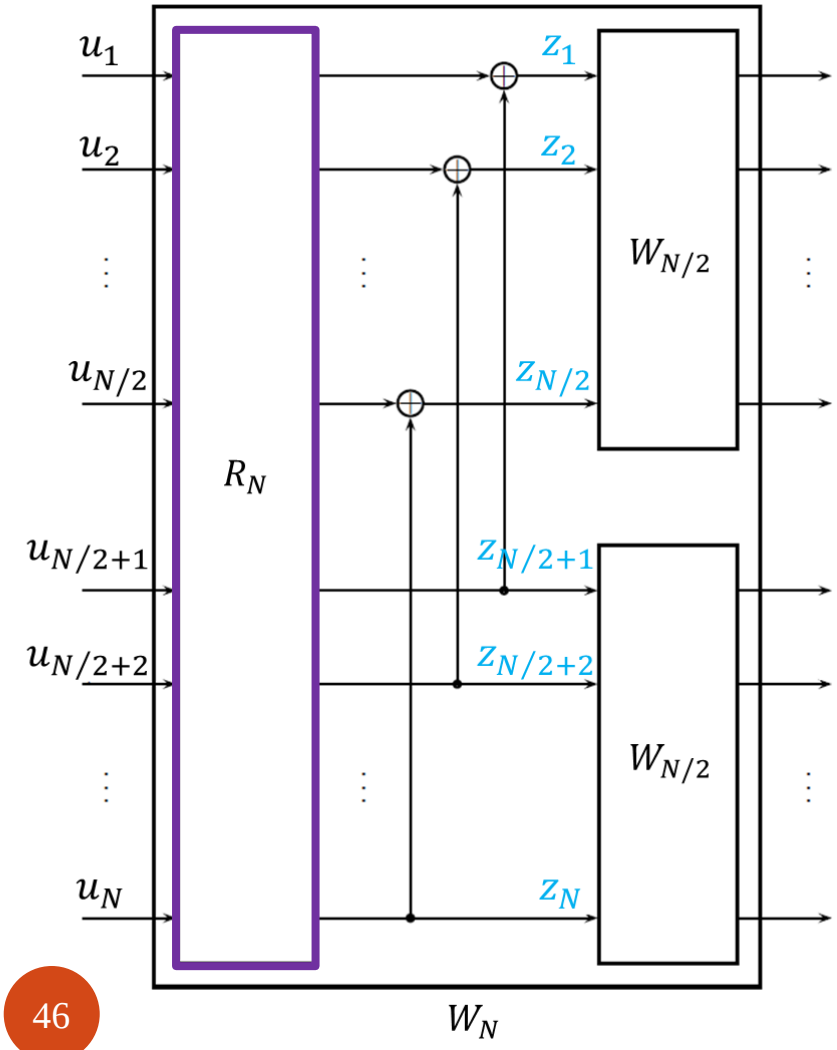
- These two figures are equivalent.
- Odd index

$$u_1 \oplus u_2 = v_1 = z_1$$
$$u_3 \oplus u_4 = v_2 = z_2$$
$$\vdots$$
$$u_{N/2-1} \oplus u_{N/2} = v_{N/2} = z_{N/2}$$

- Even index

$$u_2 = v_{N/2+1} = z_{N/2+1}$$
$$u_4 = v_{N/2+2} = z_{N/2+2}$$
$$\vdots$$
$$u_N = v_N = z_N$$

Alternative Realization for W_N



- Permutation R_N

$$(u_1, u_2, \ldots, u_{N/2}, u_{N/2+1}, \ldots, u_N) R_N = (u_1, u_3, \ldots, u_{N-1}, u_2, u_4, \ldots, u_N)$$

- Operation of z_N

$$(u_1, u_3, \ldots, u_{N-1}, u_2, u_4, \ldots, u_N) \begin{bmatrix} I_{N/2} & 0 \\ I_{N/2} & I_{N/2} \end{bmatrix} = (z_1, z_2, \ldots, z_{N/2}, z_{N/2+1}, z_{N/2+2}, \ldots, z_N)$$

where $\begin{bmatrix} I_{N/2} & 0 \\ I_{N/2} & I_{N/2} \end{bmatrix} = F \otimes I_{N/2}$.

Formulas for G_N

- Identity: $(AC) \otimes (BD) = (A \otimes B)(C \otimes D)$
- The algebraic form for G_N:

$$G_N = (I_{N/2} \otimes F) R_N (I_2 \otimes G_{N/2}), \text{ for } N \geq 2$$

with $G_1 = I_1$.

$$\Longrightarrow G_N = R_N (F \otimes I_{N/2})(I_2 \otimes G_{N/2})$$
$$= R_N (F \otimes G_{N/2})$$

Substitute $G_{N/2} = R_{N/2}(F \otimes G_{N/4})$,

$$\Longrightarrow G_N = R_N \left(F \otimes \left(R_{N/2}(F \otimes G_{N/4})\right)\right)$$
$$= R_N \left((I_2 F) \otimes \left(R_{N/2}(F \otimes G_{N/4})\right)\right)$$
$$= R_N (I_2 \otimes R_{N/2}) \left(F \otimes (F \otimes G_{N/4})\right)$$
$$= R_N (I_2 \otimes R_{N/2})(F^{\otimes 2} \otimes G_{N/4})$$

Formulas for G_N

Substitute $G_{N/4} = R_{N/4}\left(F \otimes G_{N/8}\right)$,

$$\Longrightarrow G_N = R_N\left(I_2 \otimes R_{N/2}\right)\left(F^{\otimes 2} \otimes \left(R_{N/4}\left(F \otimes G_{N/8}\right)\right)\right)$$

$$= R_N\left(I_2 \otimes R_{N/2}\right)\left(\left(I_4 F^{\otimes 2}\right) \otimes \left(R_{N/4}\left(F \otimes G_{N/8}\right)\right)\right)$$

$$= R_N\left(I_2 \otimes R_{N/2}\right)\left(I_4 \otimes R_{N/4}\right)\left(F^{\otimes 2} \otimes \left(F \otimes G_{N/8}\right)\right)$$

$$= R_N\left(I_2 \otimes R_{N/2}\right)\left(I_4 \otimes R_{N/4}\right)\left(F^{\otimes 3} \otimes G_{N/8}\right)$$

$$\vdots$$

$$= R_N\left(I_2 \otimes R_{N/2}\right)\left(I_4 \otimes R_{N/4}\right)\cdots\left(I_{N/2} \otimes R_2\right)\left(F^{\otimes n} \otimes G_{N/2^n}\right)$$

$$= R_N\left(I_2 \otimes R_{N/2}\right)\left(I_4 \otimes R_{N/4}\right)\cdots\left(I_{N/2} \otimes R_2\right)F^{\otimes n}$$

Define $B_N = R_N\left(I_2 \otimes R_{N/2}\right)\left(I_4 \otimes R_{N/4}\right)\cdots\left(I_{N/2} \otimes R_2\right)$,

$\Longrightarrow G_N = B_N F^{\otimes n}$.

Bit Reversal Operation

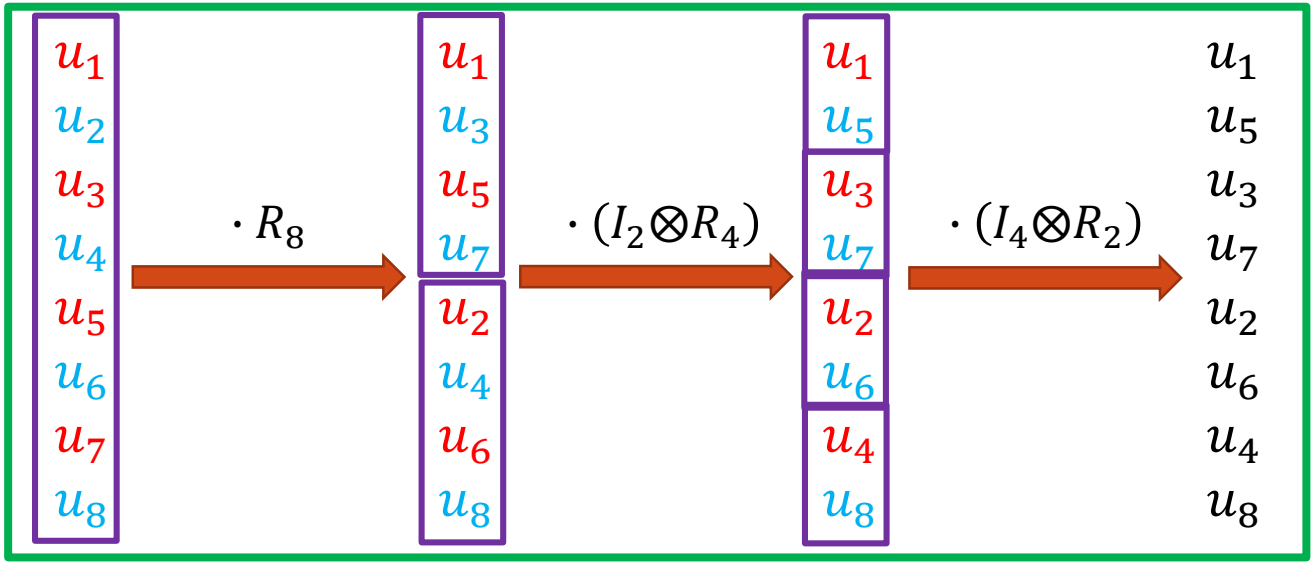
- Recall that $B_N = R_N\left(I_2 \otimes R_{N/2}\right)\left(I_4 \otimes R_{N/4}\right)\cdots\left(I_{N/2} \otimes R_2\right)$.

$$= R_N\left(I_2 \otimes B_{N/2}\right).$$

- Bit reversal for $N = 8$:

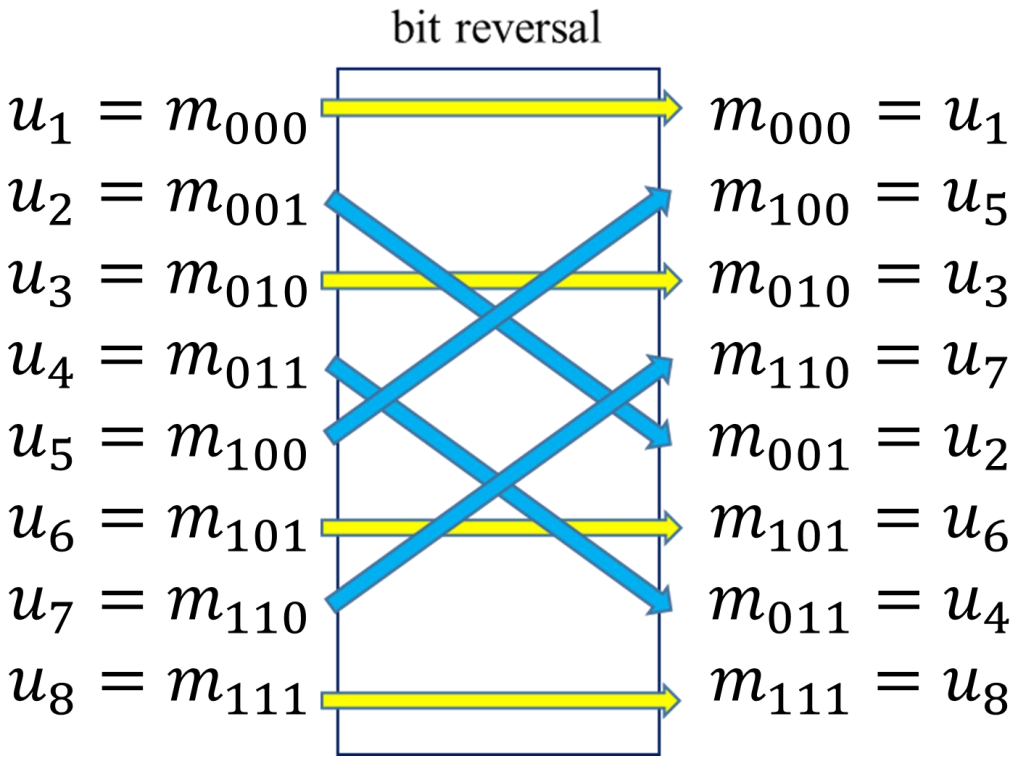
$$\Longrightarrow B_8 = R_8(I_2 \otimes R_4)(I_4 \otimes R_2)$$

$$\Longrightarrow u_1^8 B_8 = u_1^8 R_8(I_2 \otimes R_4)(I_4 \otimes R_2).$$



Alternative Realization for Bit Reversal

- Set $m_0^7 = u_1^8$, and denote the index of m with 3-digit binary form, i.e., $m_0^7 = (m_{000}, m_{001}, \cdots, m_{110}, m_{111})$.
- Equivalent operation of bit reversal:



Decoding

Decoding

- Successive cancellation (SC) decoding was proposed by Erdal Arıkan with polar code.
- The primary decoding algorithms for polar codes can be roughly divided into two schemes:
  - Successive cancellation (SC) decoding.
    - List successive cancellation (LSC) decoding.
    - CRC-aided LSC decoding.
  - Belief propagation (BP) decoding.

Decoding

- Compared with some original coding schemes such as LDPC and Turbo codes, the performance of polar codes with finite code length is limited.
- Base on the SC decoding, list successive cancellation (LSC) decoding was proposed to boost the performance of polar code.
- Instead of basing on the serial processing scheme of SC, BP decoding was proposed to improve the high decoding latency and low throughput issues of SC.

Successive Cancellation Decoding

- Consider SC decoding for an arbitrary G_N–coset code with parameter $(N,K,\mathcal{A},u_{\mathcal{A}^c})$.
- Recall that the source vector u_1^N consists of a random part $u_{\mathcal{A}}$ and a frozen part $u_{\mathcal{A}^c}$.
- The SC decoder observes $(y_1^N,u_{\mathcal{A}^c})$ and generates an estimate $\hat{u}_1^N$ of u_1^N.
- Compute the full set of LRs,

$$\left\{l_N^{(i)}\left(y_1^N,\hat{u}_1^{i-1}\right) : 1 \le i \le N\right\}.$$

Likelihood Ratio

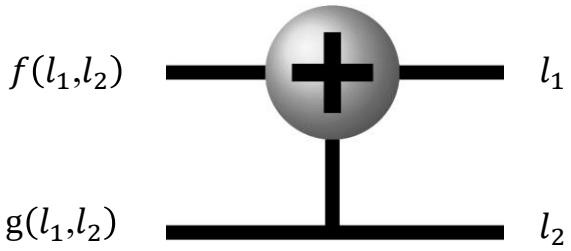
- Compute the likelihood ratio (LR):

$$l_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \triangleq \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)}$$

- Generate decision as:

$$\hat{u}_i = \begin{cases} 0, & \text{if } l_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \geq 1 \\ 1, & \text{otherwise} \end{cases}$$

Fundamental Element of Decoding Algorithm



- The upper and lower likelihood are denoted as l_1 and l_2, respectively.
- The two likelihood operations are as follows:

$$\begin{pmatrix} l_1 \\ l_2 \end{pmatrix} \rightarrow \begin{pmatrix} f(l_1,l_2) \\ g(l_1,l_2) \end{pmatrix} = \begin{pmatrix} \frac{l_1 l_2 + 1}{l_1 + l_2} \\ l_2 \cdot l_1 \text{ or } l_2 / l_1 \end{pmatrix}$$

- The operation $g(L_1, L_2)$ depends on the intermediate bit decision from the upper branch:

$$g(l_1,l_2) = \begin{cases} l_2 \cdot l_1 & \text{, if upper branch is “0”} \\ \dfrac{l_2}{l_1} & \text{, if upper branch is “1”} \end{cases}$$

Recursive Formulas

$$l_N^{(2i-1)}\left(y_1^N, \hat{u}_1^{2i-2}\right) = \frac{l_{N/2}^{(i)}\left(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) l_{N/2}^{(i)}\left(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2}\right) + 1}{l_{N/2}^{(i)}\left(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) + l_{N/2}^{(i)}\left(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2}\right)}$$

$$l_N^{(2i)}\left(y_1^N, \hat{u}_1^{2i-1}\right) = \left[l_{N/2}^{(i)}\left(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right)\right]^{1-2\hat{u}_{2i-1}} \cdot l_{N/2}^{(i)}\left(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2}\right)$$

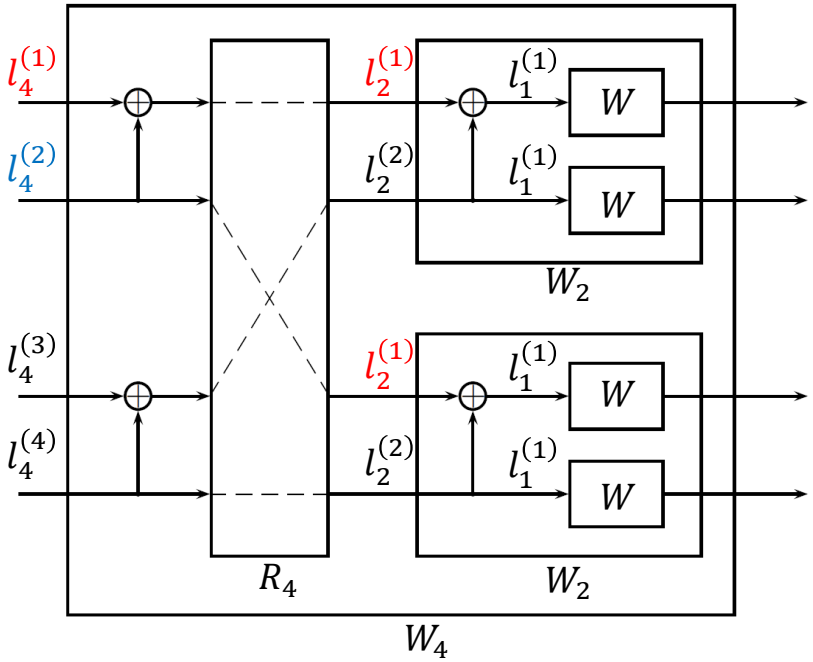
- The calculation of a LR at length N is reduced to the calculation of two LRs at length $N/2$.
- The recursion can be continued down to block length 1, i.e., $l_1^{(1)}(y_i) = \frac{W(y_i|0)}{W(y_i|1)}$.

From $N = 4$ to $N = 2$

- For $N = 4$, $i = 1$:

$$l_4^{(1)}(y_1^4) = \frac{l_2^{(1)}(y_1^2)\, l_2^{(1)}(y_3^4) + 1}{l_2^{(1)}(y_1^2) + l_2^{(1)}(y_3^4)}$$

$$l_4^{(2)}(y_1^4, \hat{u}_1) = \left[l_2^{(1)}(y_1^2)\right]^{1-2\hat{u}_1} \cdot l_2^{(1)}(y_3^4)$$

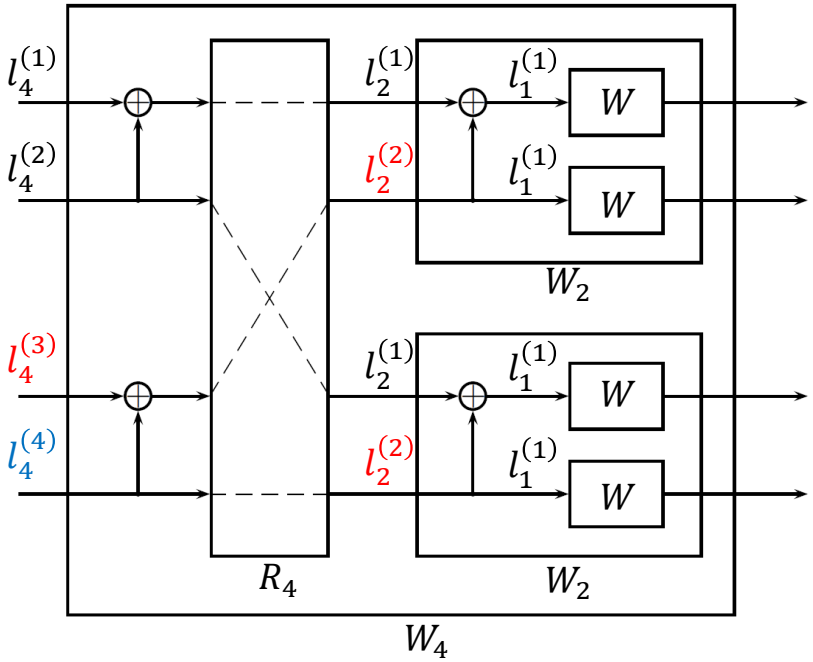


From $N = 4$ to $N = 2$

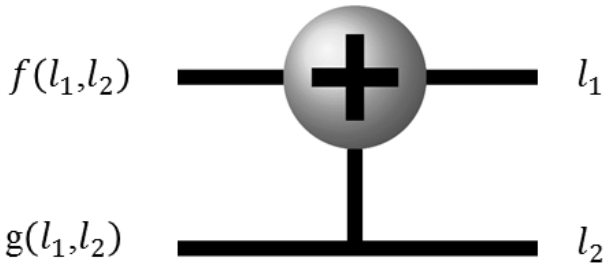
- For $N = 4$, $i = 2$:

$$l_4^{(3)}(y_1^4, \hat{u}_1^2) = \frac{l_2^{(2)}(y_1^2, \hat{u}_1 \oplus \hat{u}_2) l_2^{(2)}(y_3^4, \hat{u}_2) + 1}{l_2^{(2)}(y_1^2, \hat{u}_1 \oplus \hat{u}_2) + l_2^{(2)}(y_3^4, \hat{u}_2)}$$

$$l_4^{(4)}(y_1^4, \hat{u}_1^3) = \left[l_2^{(2)}(y_1^2, \hat{u}_1 \oplus \hat{u}_2) \right]^{1-2\hat{u}_3} \cdot l_2^{(2)}(y_3^4, \hat{u}_2)$$



Numerical Issue



- Note that

$$\begin{pmatrix} l_1 \\ l_2 \end{pmatrix} \rightarrow \begin{pmatrix} f(l_1,l_2) \\ g(l_1,l_2) \end{pmatrix} = \begin{pmatrix} \frac{l_1 l_2 + 1}{l_1 + l_2} \\ l_2 \cdot l_1 \text{ or } l_2 / l_1 \end{pmatrix}$$

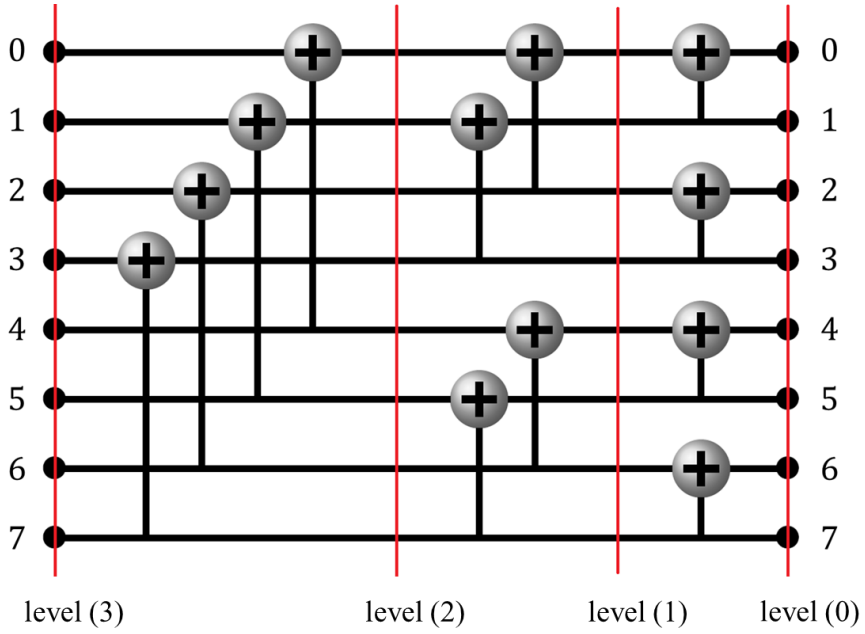
- Due to the numerical issue, we use LLRs instead of LRs:

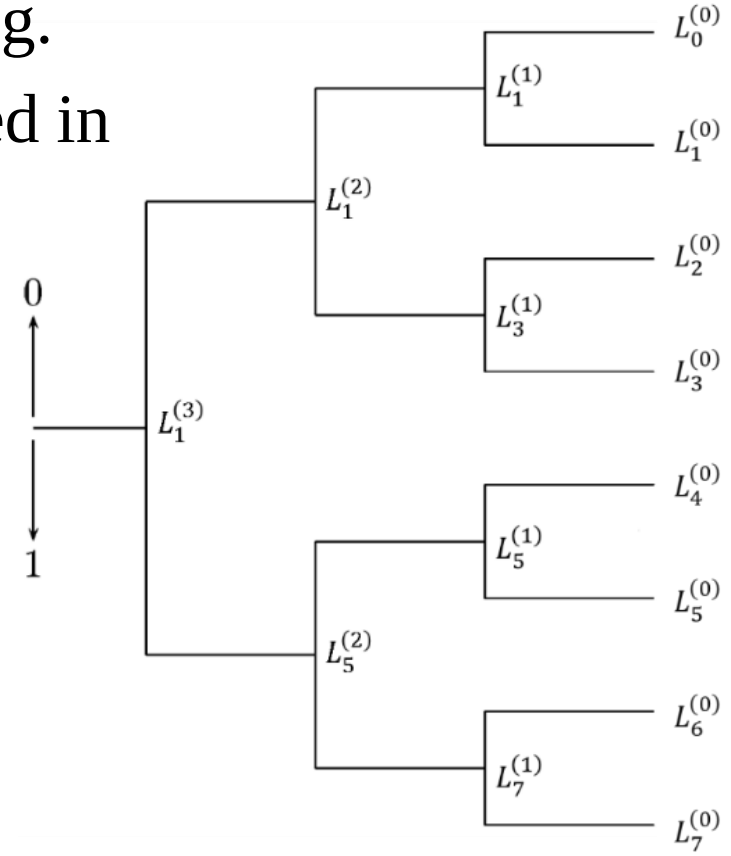
$$\begin{pmatrix} L_1 \\ L_2 \end{pmatrix} \rightarrow \begin{pmatrix} \ln(l_1) \\ \ln(l_2) \end{pmatrix} \rightarrow \begin{pmatrix} \ln f(e^{L_1}, e^{L_2}) \\ \ln g(e^{L_1}, e^{L_2}) \end{pmatrix} = \begin{pmatrix} \ln \frac{e^{L_1 + L_2} + 1}{e^{L_1} + e^{L_2}} \\ L_2 + L_1 \text{ or } L_2 - L_1 \end{pmatrix}$$

$$\approx \begin{pmatrix} \text{sign}(L_1)\text{sign}(L_2)\min\{|L_1|, |L_2|\} \\ L_2 + (-1)^u L_1 \end{pmatrix}$$

Computational Tree

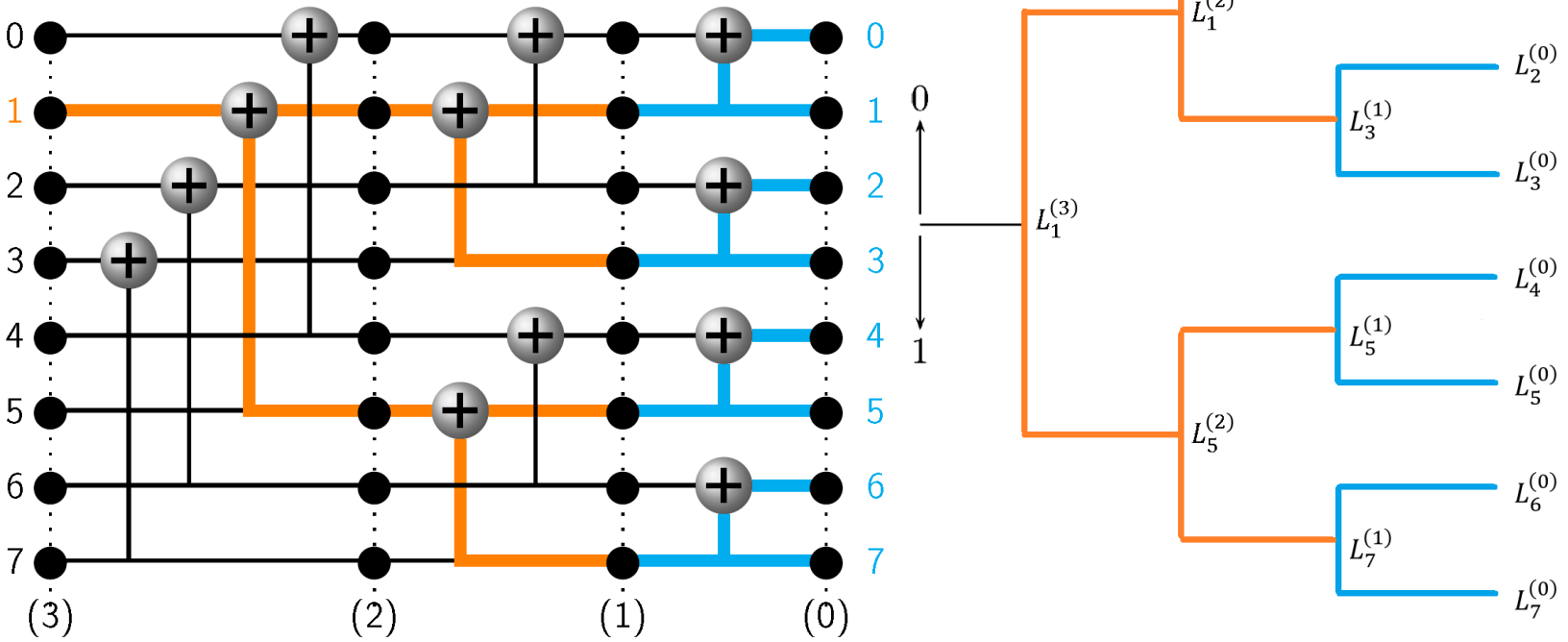
- Example with $N = 8$.
- Each bit has a tree for decoding.
- These N trees can be embedded in an $N \times (n+1)$ array.





Computational Tree

- Each bit has a tree for decoding.

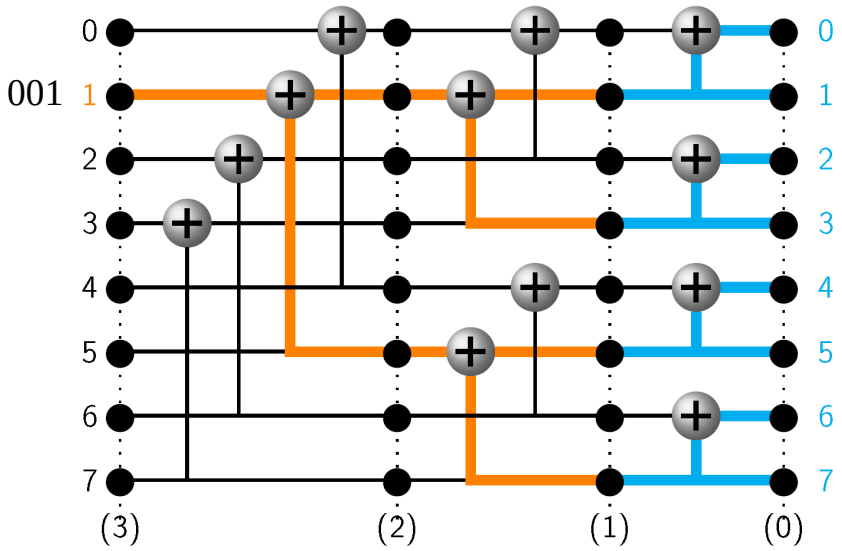


Active Level

- Depth of active-level rooted at i

$$= \begin{cases} 3 & , \text{for } i = 0 \\ 1 + (\text{number of consecutive zeros in binary} - i \text{ counted from MSB}) & , \text{for } i > 0 \end{cases}$$

- Depth of active-level rooted at $1 = 1 + 2 = 3$.

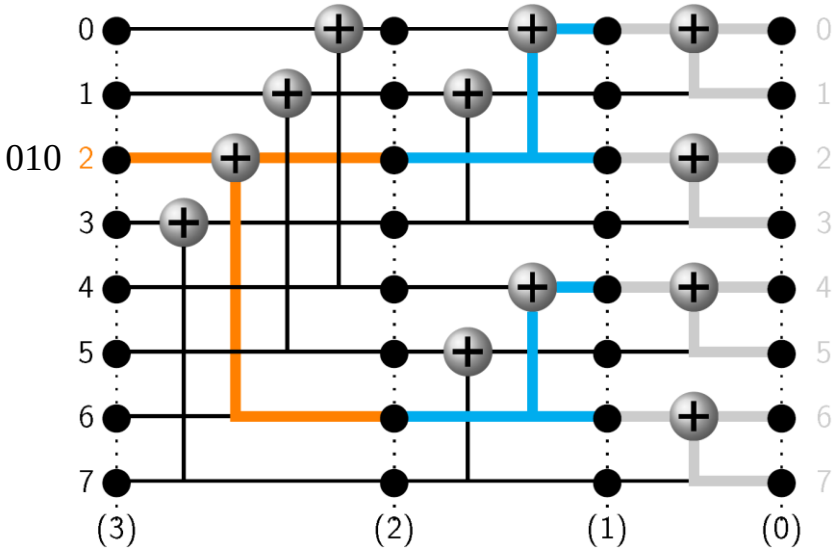


Active Level

- Depth of active-level rooted at i

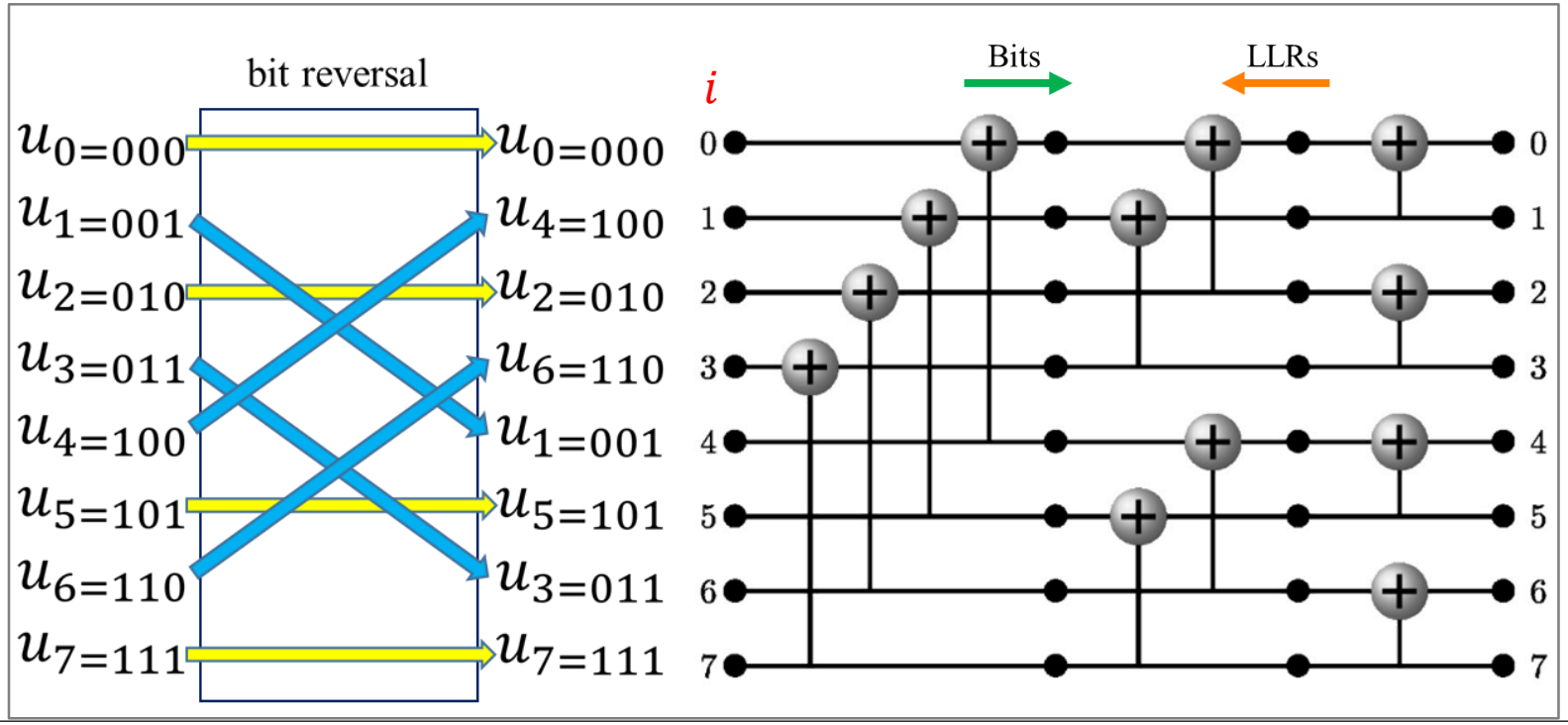
$$= \begin{cases} 3 & , \text{for } i = 0 \\ 1 + (\text{number of consecutive zeros in binary} - i \text{ counted from MSB}) & , \text{for } i > 0 \end{cases}$$

- Depth of active-level rooted at $2 = 1 + 1 = 2$.



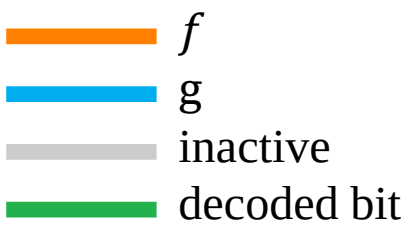
SC Decoding

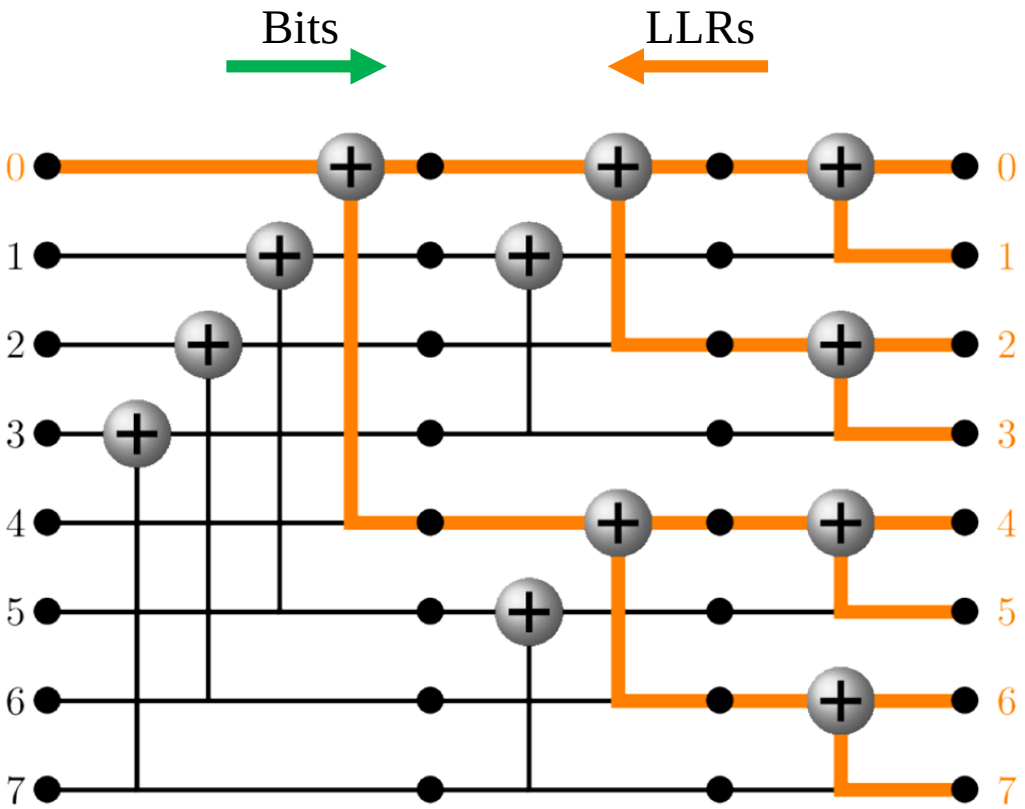
- Recall that $x_1^N = u_1^N B_N F^{\otimes n}$.
- By the active-level rule, the decoding order of i is 0, 4, 2, 6, 1, 5, 3, 7.
- The decoding order is $u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7$.
- The SC decoding algorithm decodes the bit "successively".



SC Decoding

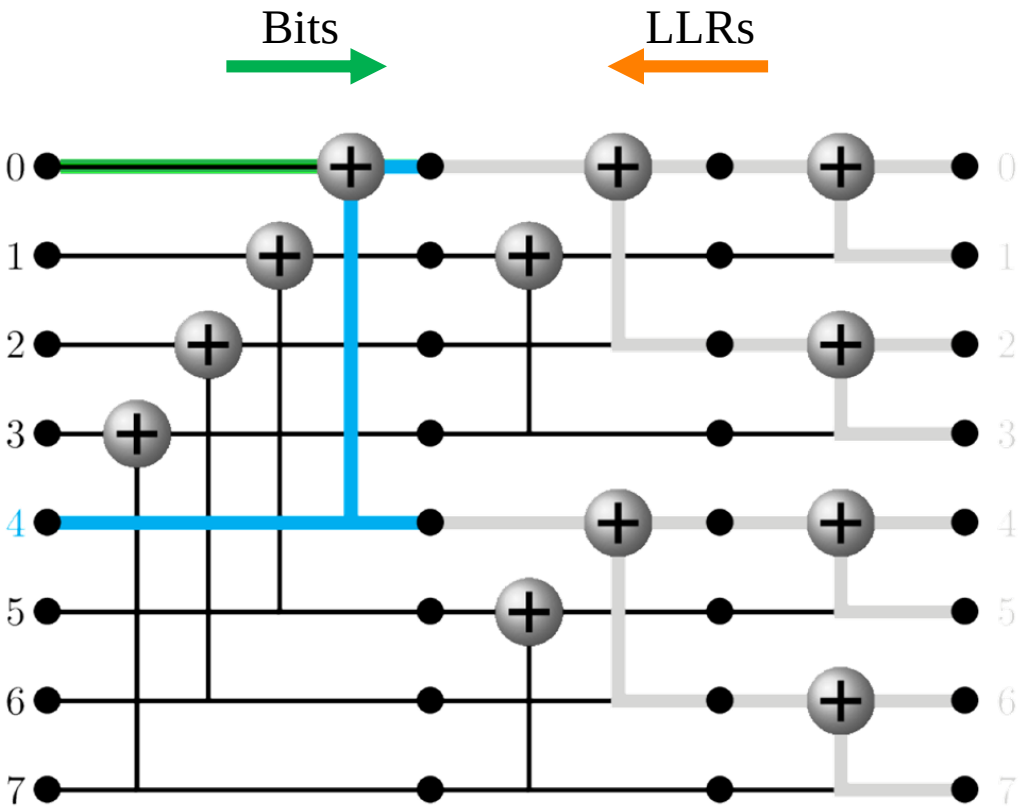
- Recall that the decoding order of i is 0, 4, 2, 6, 1, 5, 3, 7.
- Depth of active-level rooted at $0 = 3$.





SC Decoding

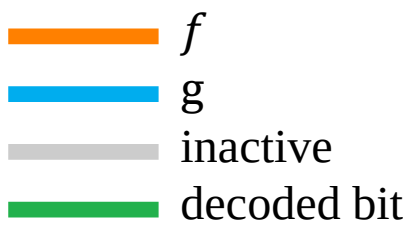
- Recall that the decoding order of i is 0, 4, 2, 6, 1, 5, 3, 7.
- Depth of active-level rooted at $4 = 1 + 0 = 1$.

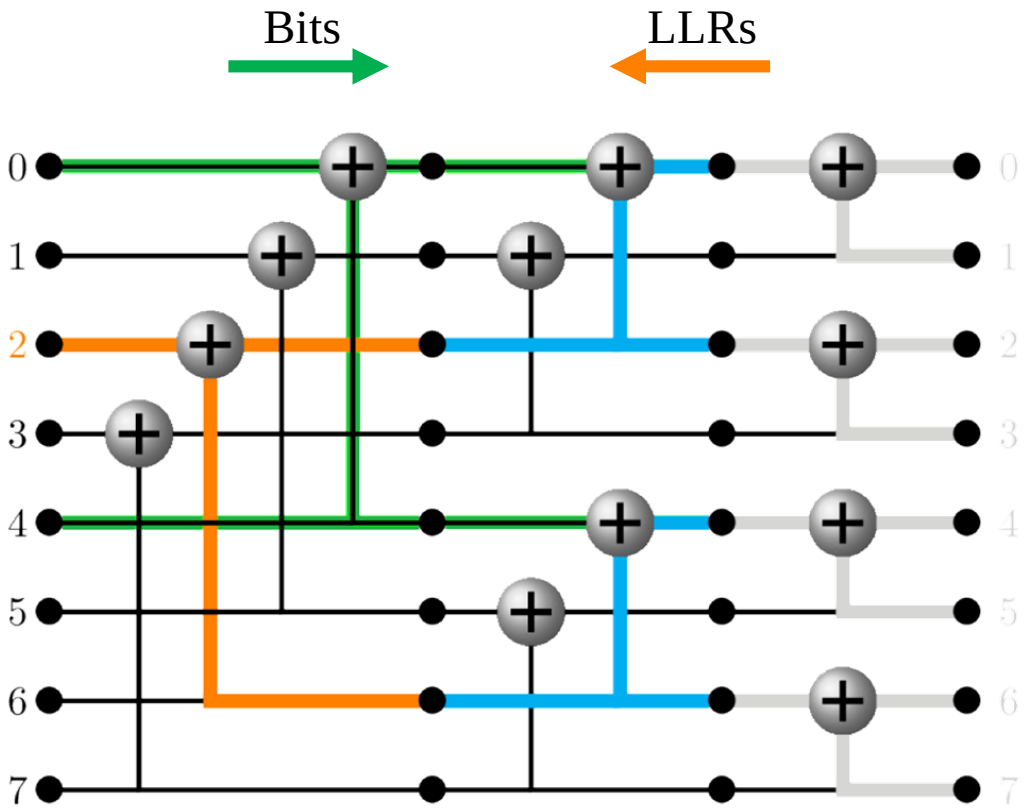


f
g
inactive
decoded bit

SC Decoding

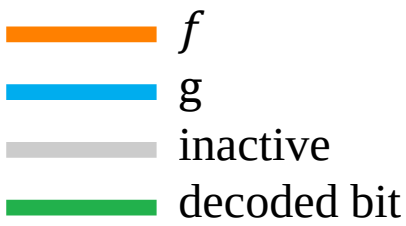
- Recall that the decoding order of i is 0, 4, 2, 6, 1, 5, 3, 7.
- Depth of active-level rooted at $2 = 1 + 1 = 2$.

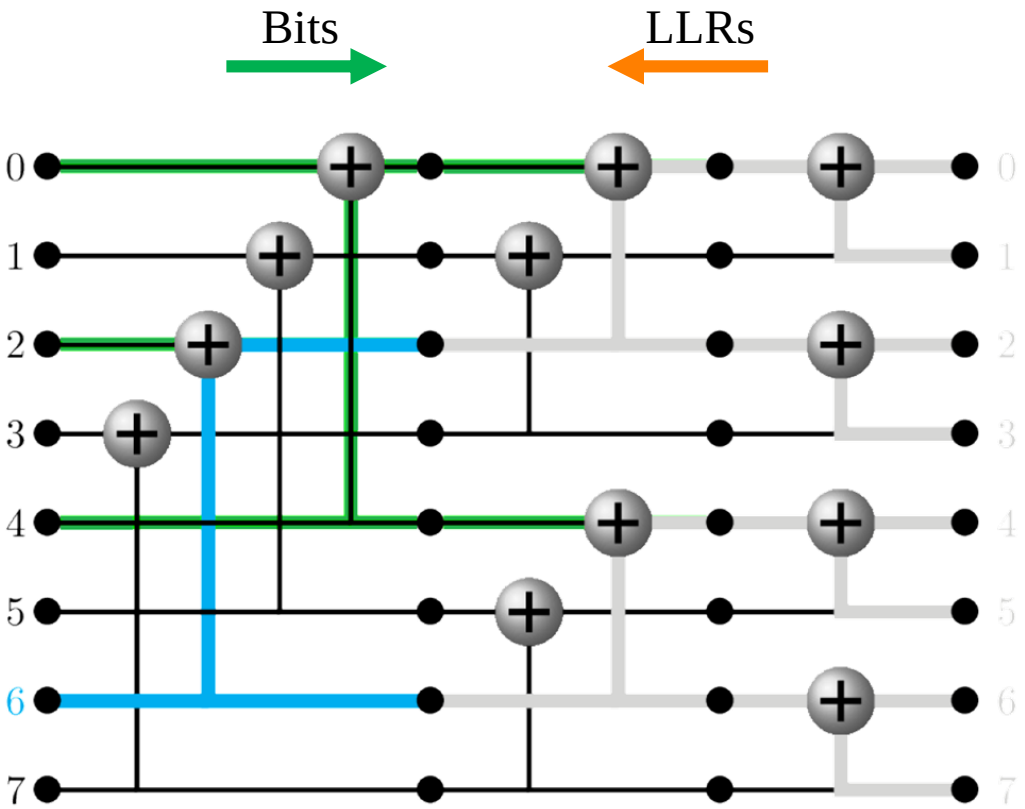




SC Decoding

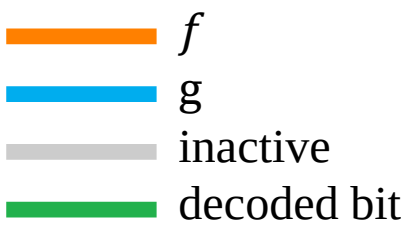
- Recall that the decoding order of i is 0, 4, 2, 6, 1, 5, 3, 7.
- Depth of active-level rooted at $6 = 1 + 0 = 1$

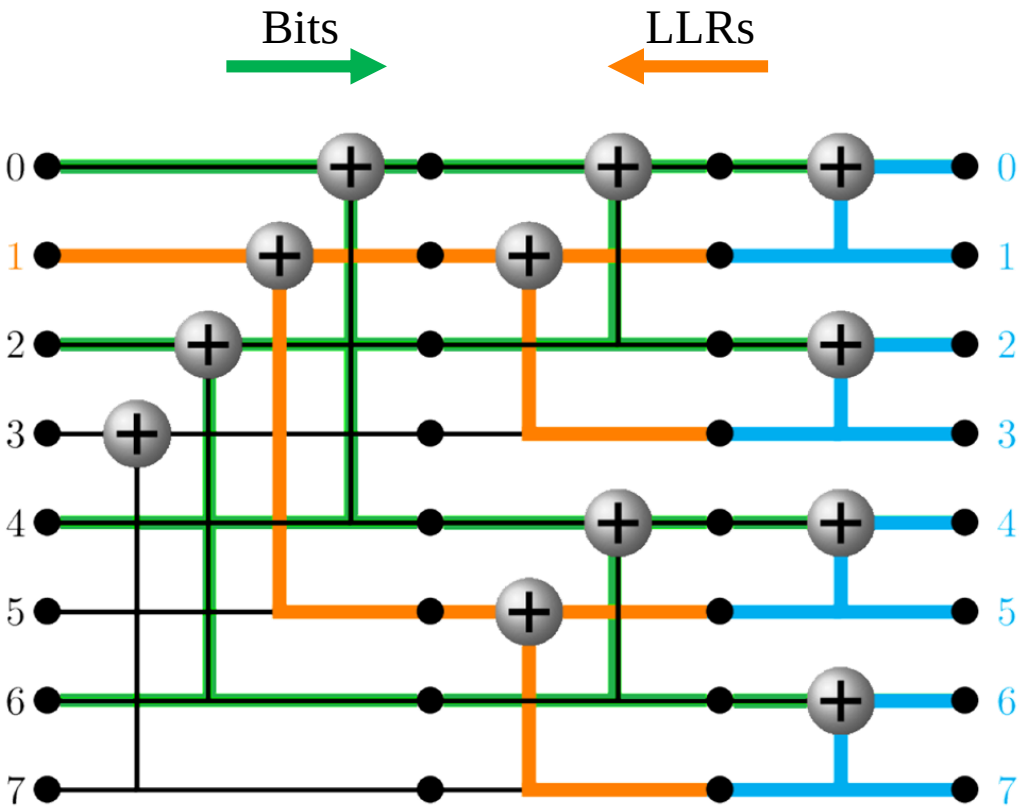




SC Decoding

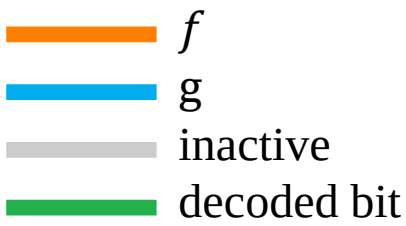
- Recall that the decoding order of i is 0, 4, 2, 6, 1, 5, 3, 7.
- Depth of active-level rooted at $1 = 1 + 2 = 3$

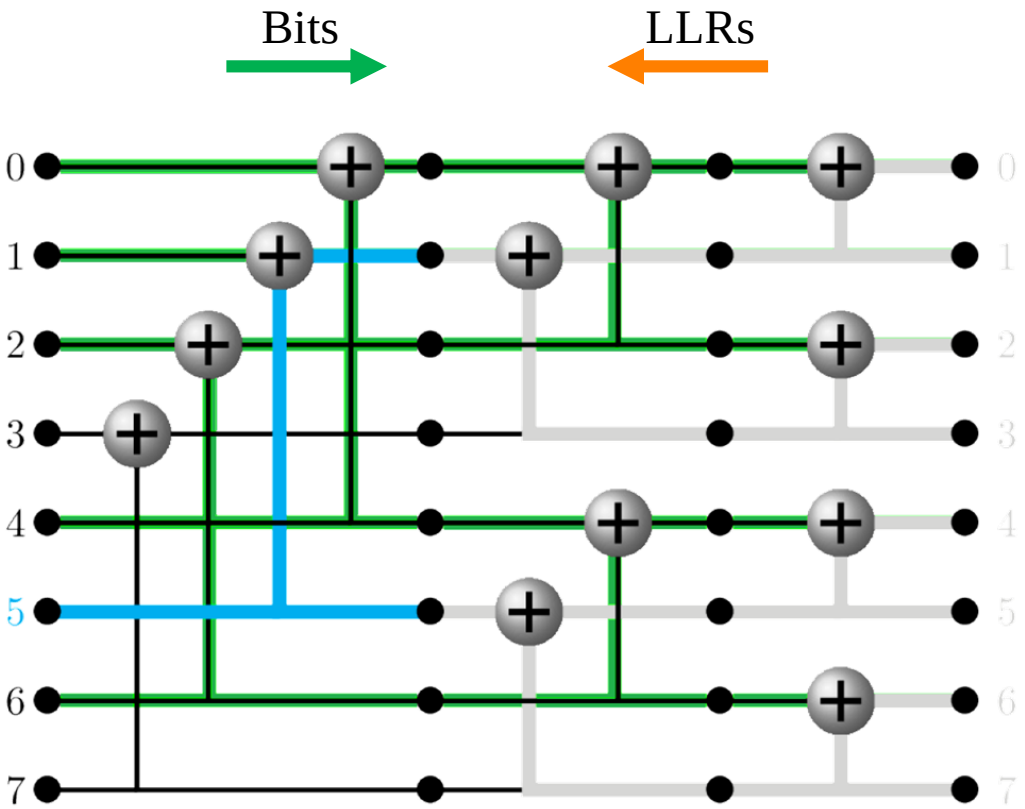




SC Decoding

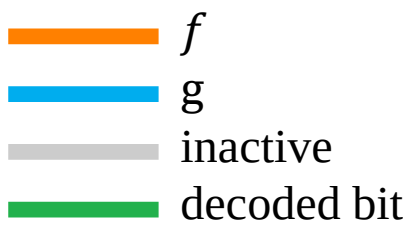
- Recall that the decoding order of i is 0, 4, 2, 6, 1, 5, 3, 7.
- Depth of active-level rooted at $5 = 1 + 0 = 1$

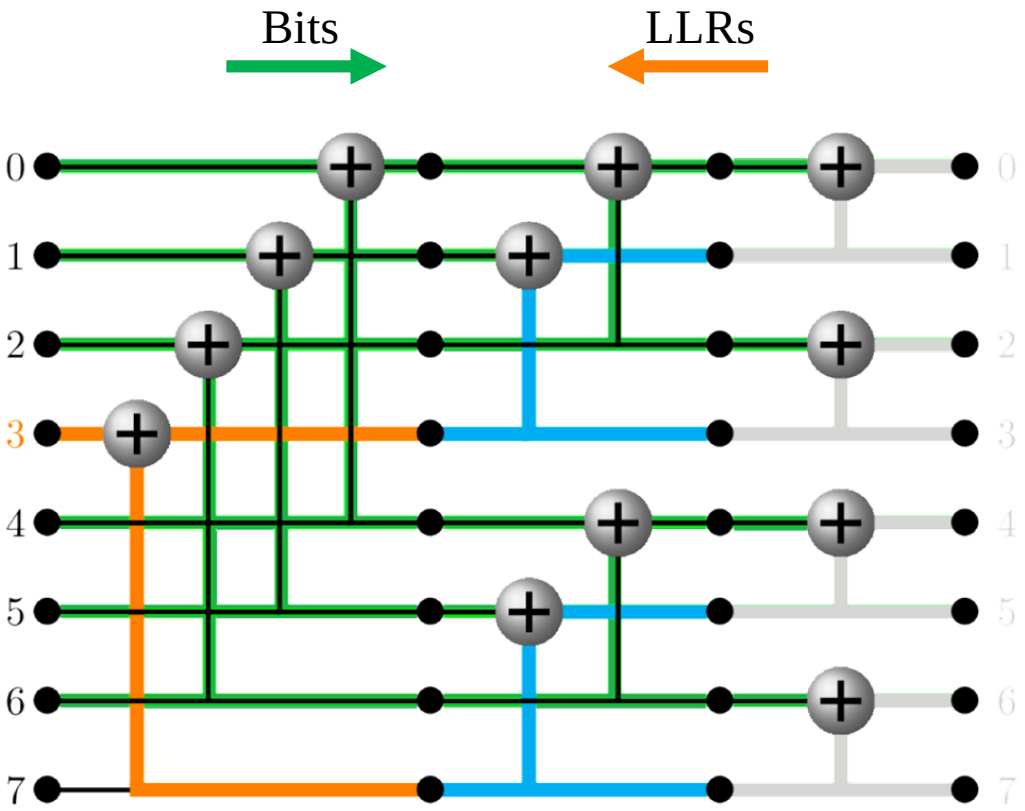




SC Decoding

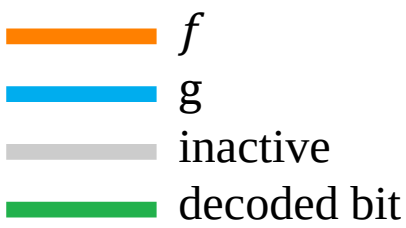
- Recall that the decoding order of i is 0, 4, 2, 6, 1, 5, 3, 7.
- Depth of active-level rooted at $3 = 1 + 1 = 2$

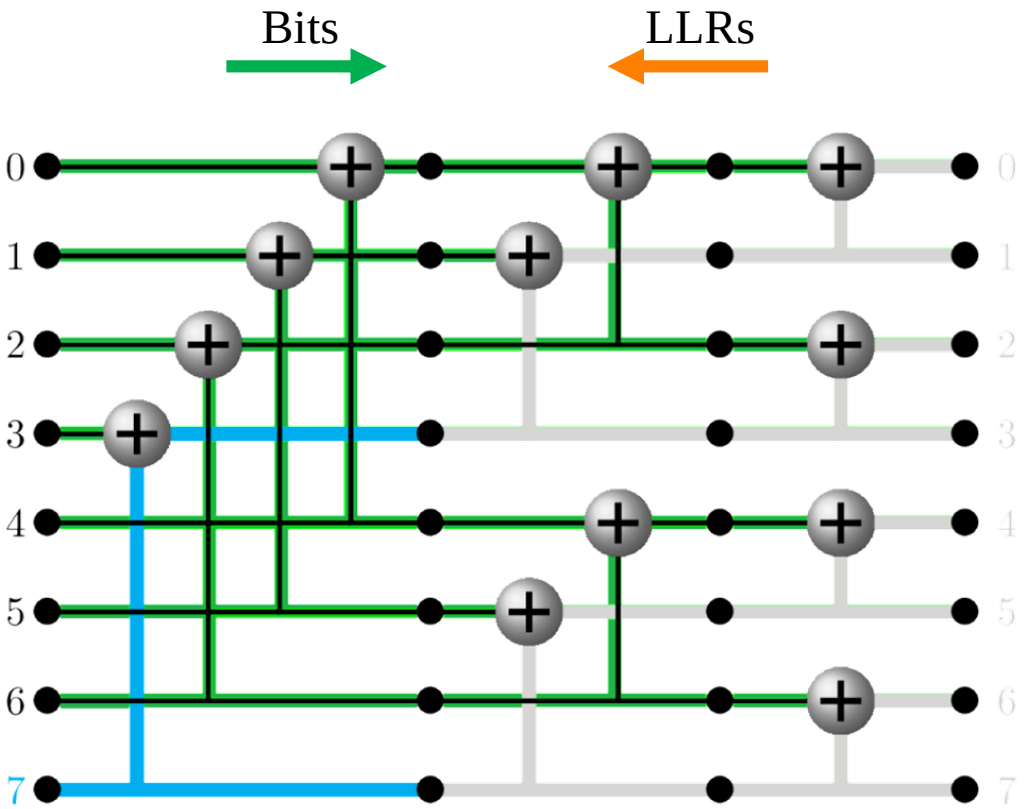




SC Decoding

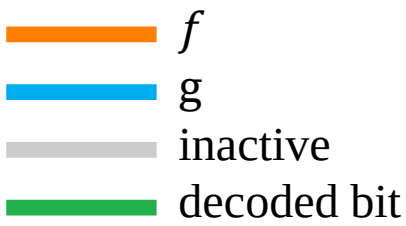
- Recall that the decoding order of i is 0, 4, 2, 6, 1, 5, 3, 7.
- Depth of active-level rooted at $7 = 1 + 0 = 1$

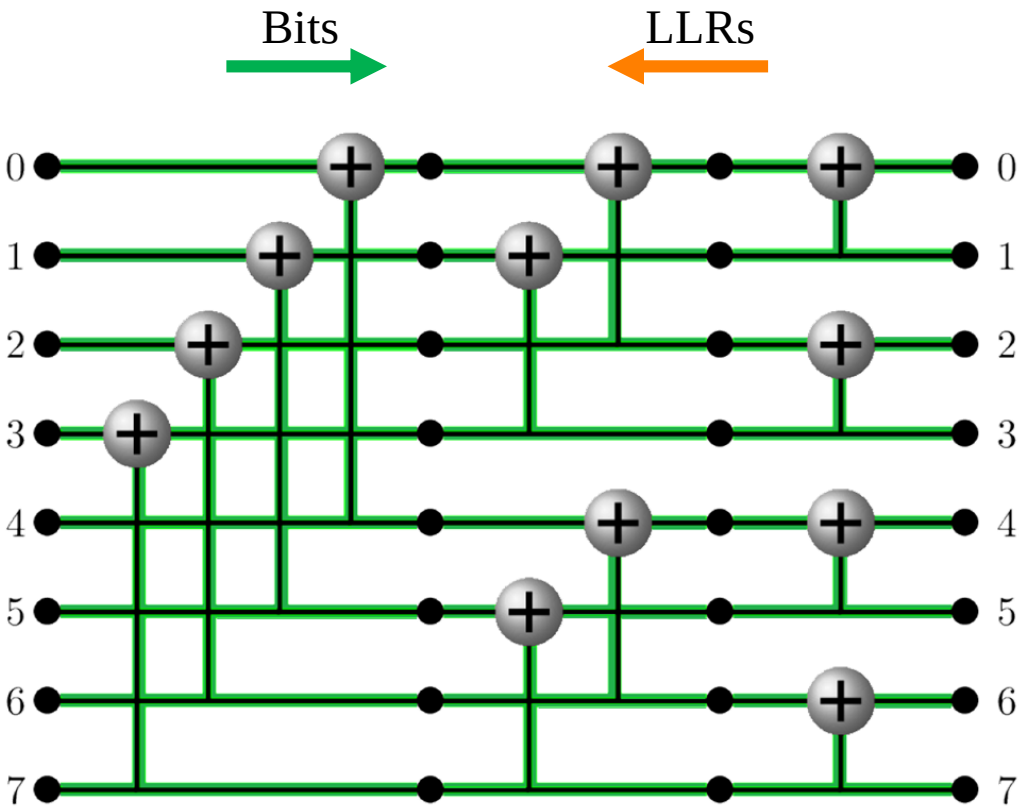




SC Decoding

- Recall that the decoding order of i is 0, 4, 2, 6, 1, 5, 3, 7.
- Rearrange the decoded bits by bit-reversal to get the sequence u_0^7.





List Successive Cancellation Decoding

- LSC decoding simultaneously produces at most L locally best candidates during the decoding process to reduce the chance of missing the correct codeword.
- Notice that the traditional SC decoding can be regarded as an LSC decoding with $L = 1$.

Recall for Successive Cancellation Decoding

- For SC decoding, if a bit $\hat{u}_i$ is frozen, then $\hat{u}_i = 0$. Otherwise, the decoding rule is as follows:

$$\hat{u}_i = \begin{cases} 0, & \dfrac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)} \geq 1 \\ 1, & \text{otherwise} \end{cases}$$

- It can be found that once a wrong bit decision is made, there is no chance to correct it in the following decoding procedure.

List Successive Cancellation Decoding

- LSC allows more than one partial sequence to be further processed after each step of SC.
- The L most probable partial sequences are recorded and updated in a size-L list according to their probabilities.

LSC Decoding Algorithm

- Notations and variables:
  - Let $S^{(i)}$ denote the set of candidate sequences in the ith step of the decoding process.
  - Let $\left|S^{(i)}\right|$ be the size of $S^{(i)}$.
  - Set L to be the maximum allowed size of the list.

1. Initialization: Set a null sequence as the only candidate in the initial list and set the corresponding probability to be 1, i.e., $S^{(0)} = \{\emptyset\}$, $P(\emptyset) = 1$.

LSC Decoding Algorithm

2. Bit estimation: Bits are estimated successively with index $i = 1, 2, \cdots, N$:
   a) Expansion: For each candidate in the list, generate two length-i sequences with decoding $\hat{u}_i$ as "0" and "1", respectively:

$$S^{(i)} = \left\{ \hat{u}_1^i \middle| \hat{u}_1^{i-1} \in S^{(i-1)}, \hat{u}_i \in \{0,1\} \right\}$$

For each $\hat{u}_1^i \in S^{(i)}$, calculate its probability:

$$P\left(\hat{u}_1^i\right) = P\left(\hat{u}_1^{i-1}\right) P\left(\hat{u}_i = b \middle| u_1^{i-1} = \hat{u}_1^{i-1}\right)$$

where $b \in \{0,1\}$.

LSC Decoding Algorithm

The bit conditional probability is defined as follows:

$$P\left(\hat{u}_i = b \middle| u_1^{i-1} = \hat{u}_1^{i-1}\right)$$

$$= \begin{cases} \dfrac{W_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1} \middle| \hat{u}_i = b\right)}{\sum_{b' \in \{0,1\}} W_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1} \middle| \hat{u}_i = b'\right)} & \text{, if } \hat{u}_i \text{ is info. bit} \\ \mathbf{1}_{b=0} & \text{, if } \hat{u}_i \text{ is froz. bit} \end{cases}$$

where $\mathbf{1}_c$ is the indicator function equal to 1 if condition c is fulfilled.

The size of candidate set is doubled after this phase, i.e. $\left|S^{(i)}\right| = 2\left|S^{(i-1)}\right|$.

LSC Decoding Algorithm

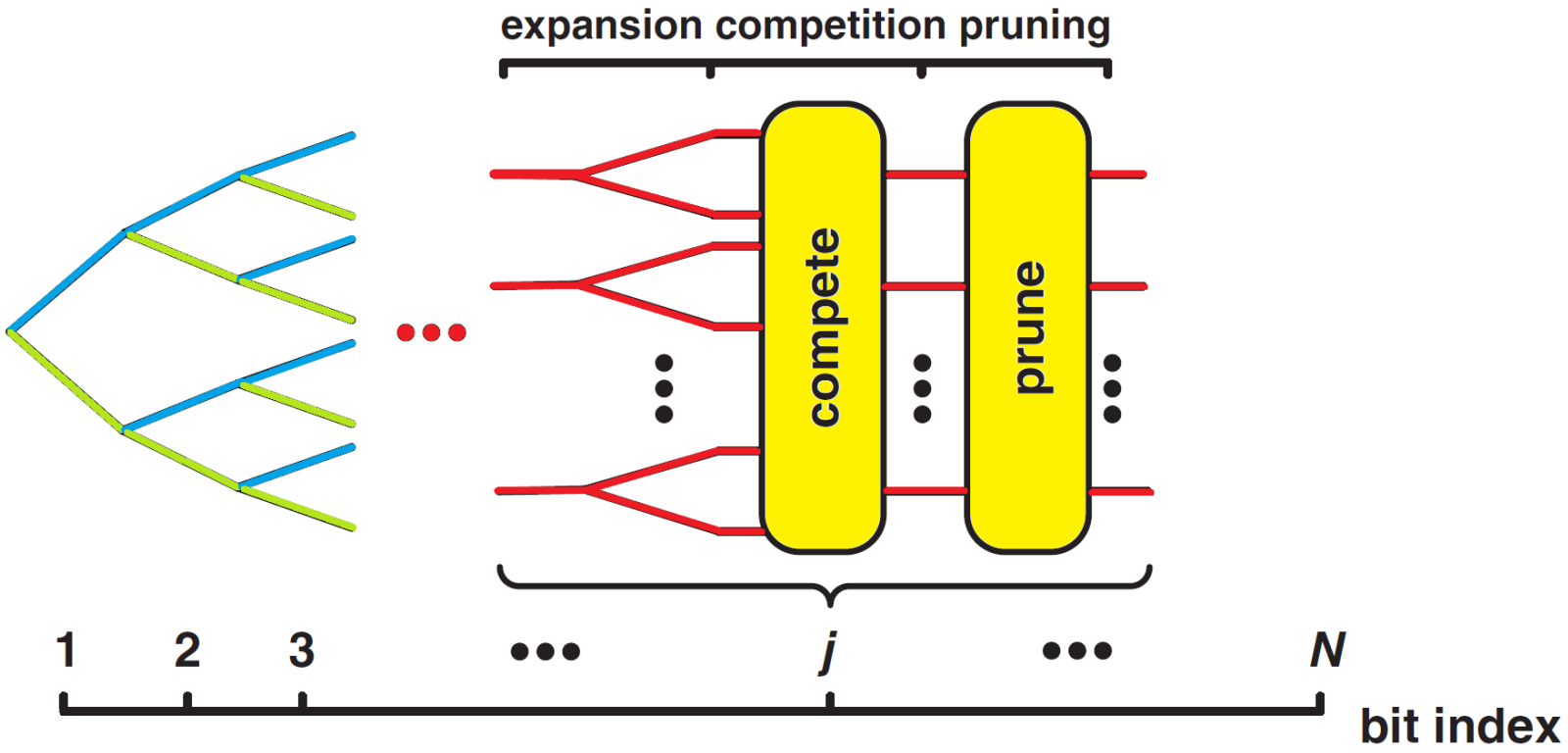
b) Competition: If the number of candidates $\left|S^{(i)}\right|$ after 2a) is not larger than L, then skip this step; otherwise, reserve L candidates with the largest probabilities and drop the others from $S^{(i)}$.

3. Sequence determination: After the N bits are decoded, re-encode every candidate in the list and calculate corresponding likelihood probabilities. Select the one having the maximal probability as the sequence estimation:

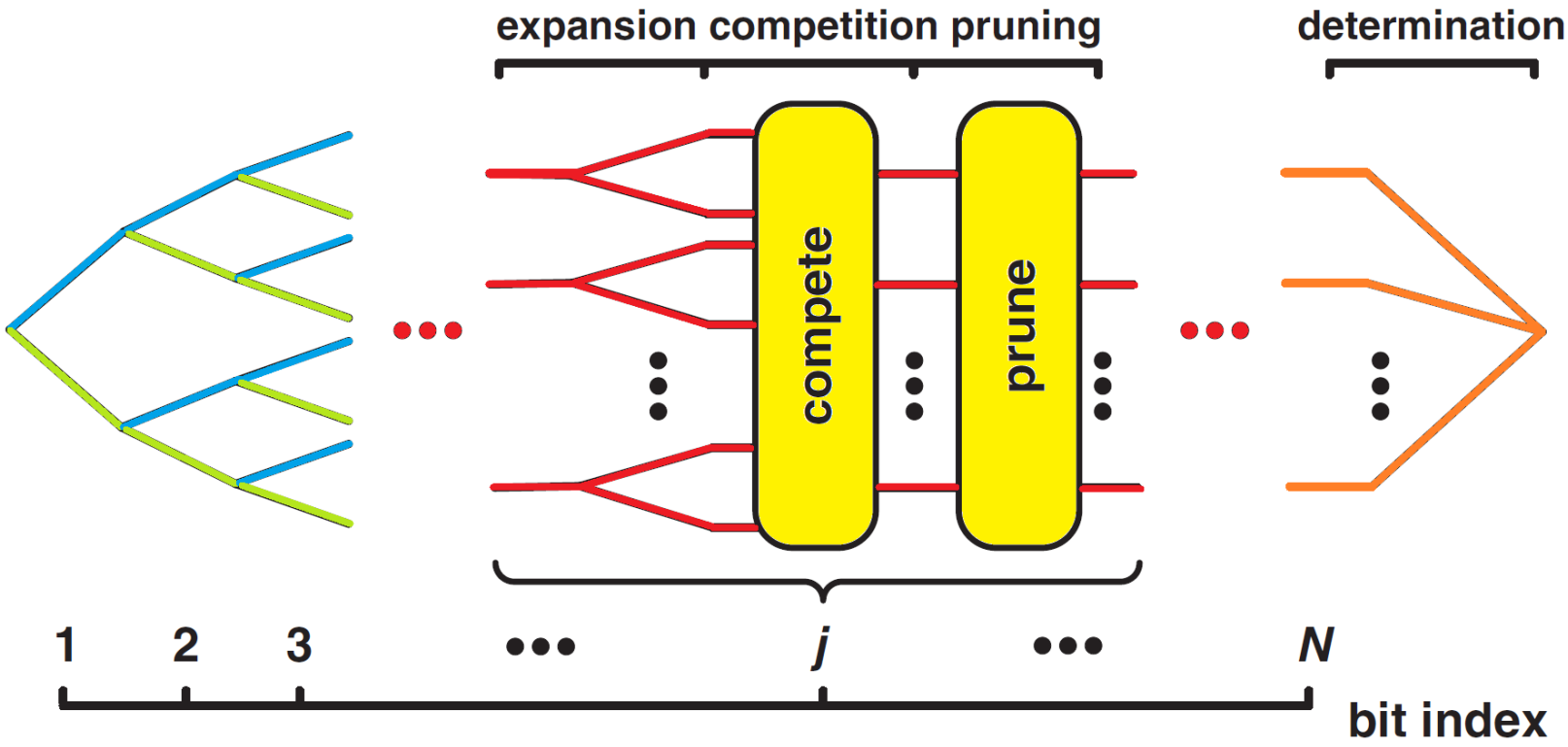
$$\hat{u}_1^N = \arg \max_{\hat{v}_1^N \in S^{(N)}} \prod_{i=1}^{N} W\left(y_i \middle| x_i = (\hat{v}_1^N \cdot G_N)_i\right)$$

Decoding Procedure

Decoding Procedure



Decoding Procedure

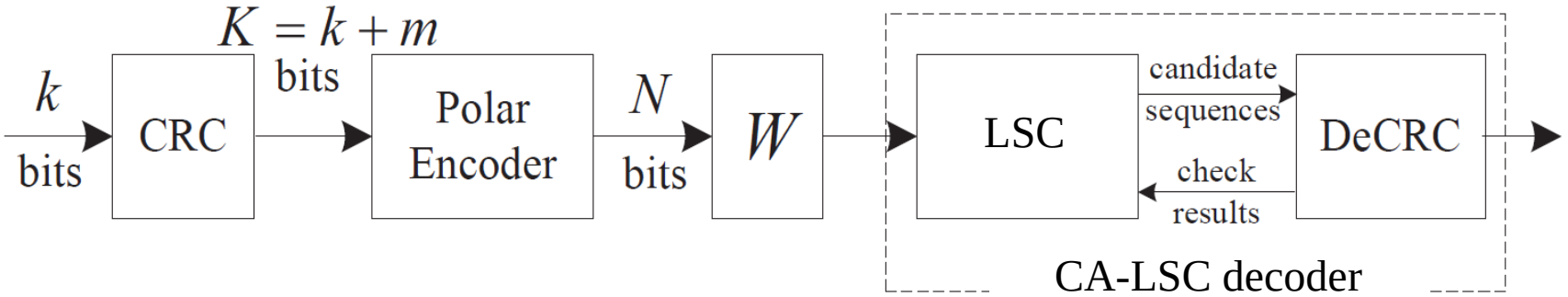


CRC-aided LSC Decoding

- CRC provides a stopping criterion for the iterative decoding process or starts retransmission requests by utilizing the checking information provided by CRC detector in a codeword selection mechanism.
- CRC-aided path selection: After N bits are decoded, the paths in the list are examined one-by-one with decreasing reliability. The decoder outputs the first path passing the CRC detection as the estimation sequence. If none of such a path in $S^{(i)}$ can pass, then decoding declares a decoding failure.

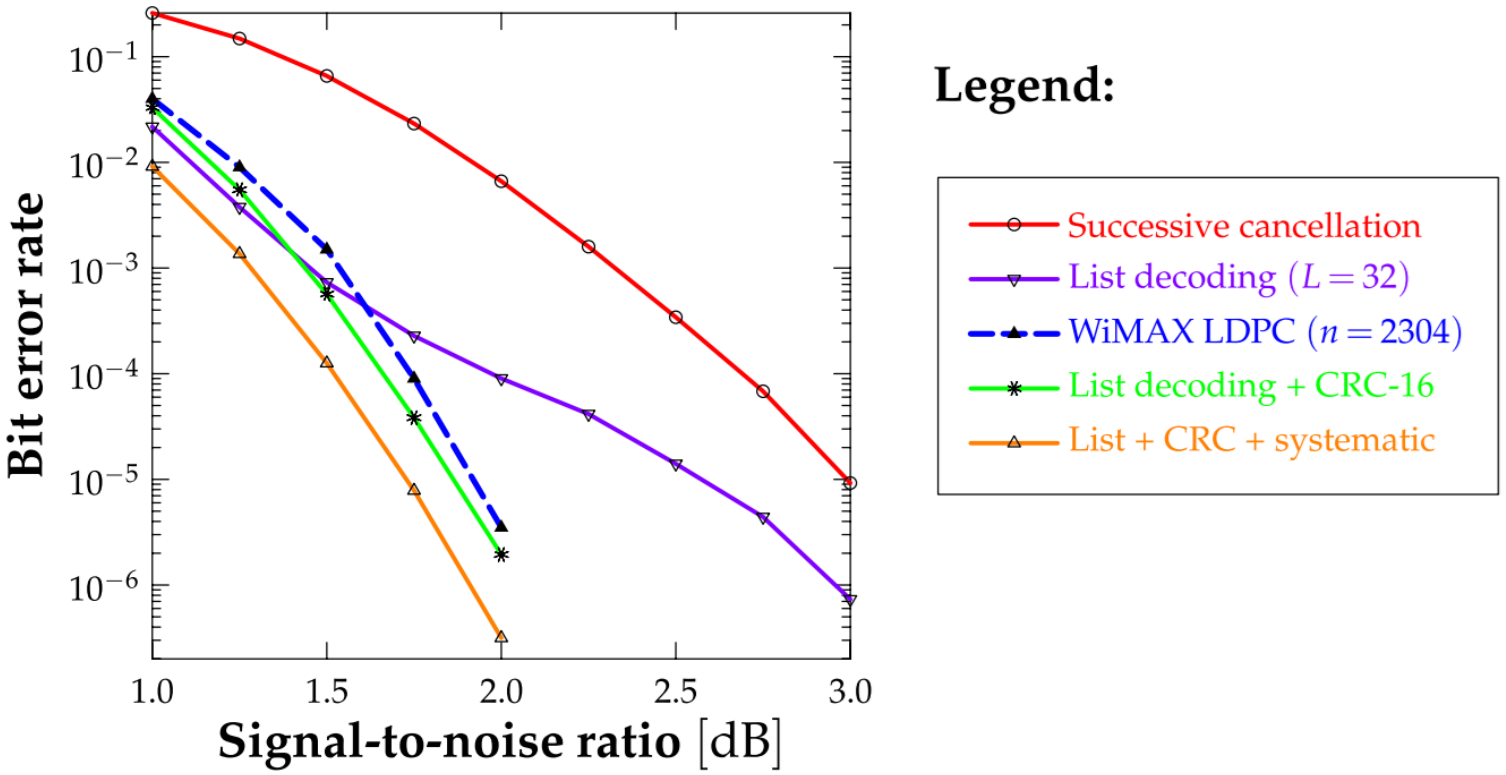
CRC-aided LSC Decoding

- CRC-aided polar code schemes.



Simulations

- Simulation over the BPSK-AWGN channel for polar codes of length $n = 2048$ with rate 0.5.

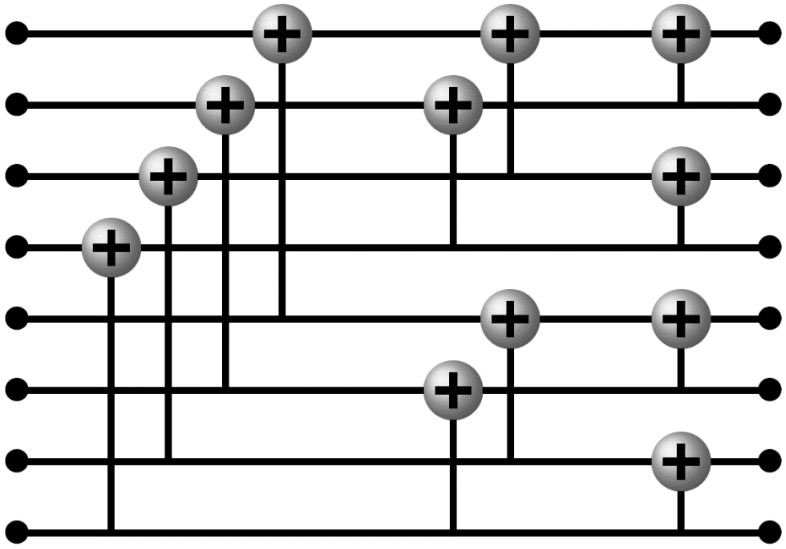


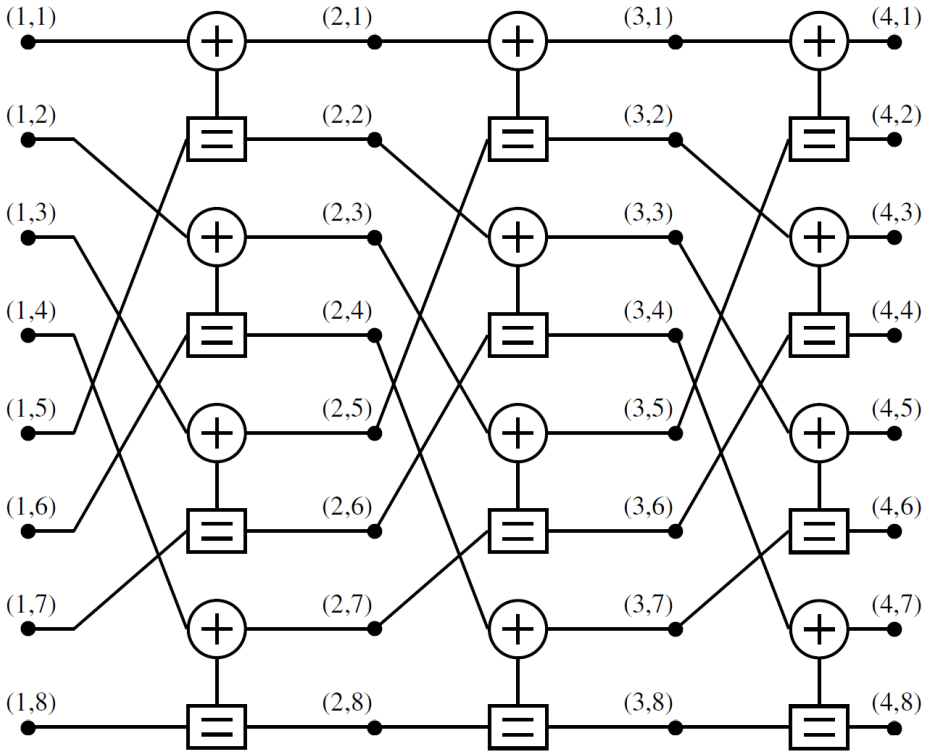
Belief Propagation Decoding

- Instead of basing on the serial processing scheme as SC decoding, BP decoding has much more parallel scheme than SC decoding.
- BP decoding improves the high decoding latency and low throughput issues of SC.

Uniform Factor Graph Representation

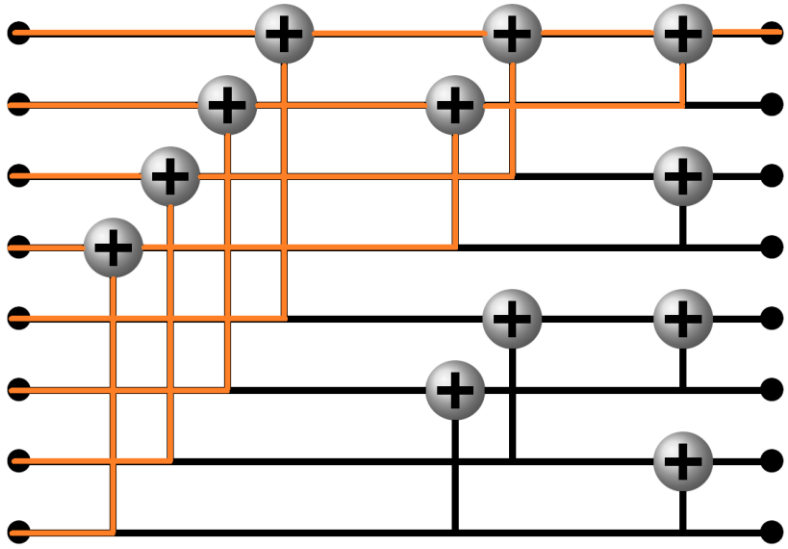
- The following two graphs do exactly the same thing.
- Remind that u_1^8 should be operated by bit-reversal before input to the graph to complete the encoding.

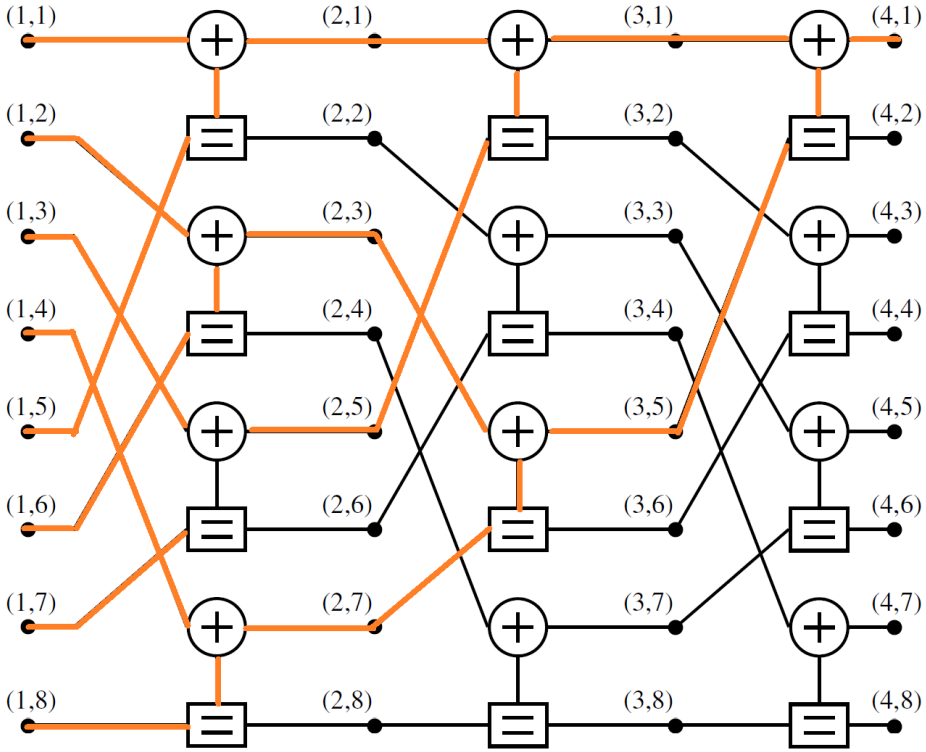




Uniform Factor Graph Representation

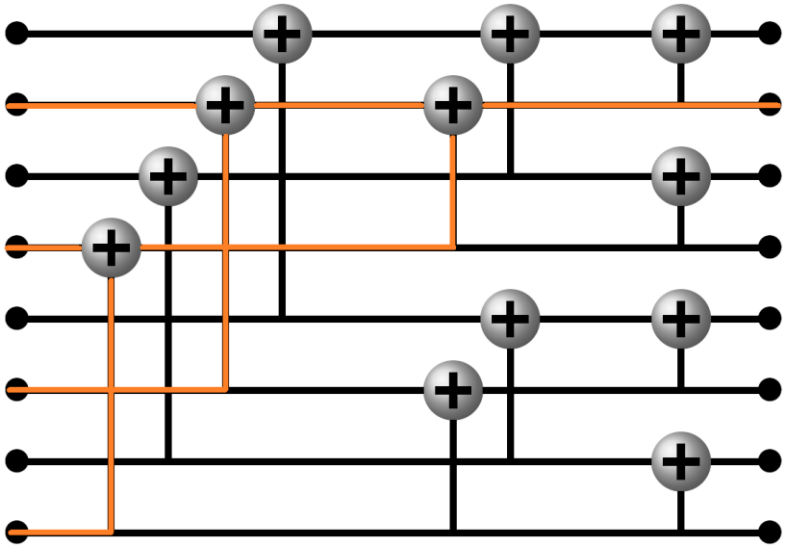
- The following two graphs do exactly the same thing.
- Remind that u_1^8 should be operated by bit-reversal before input to the graph to complete the encoding.

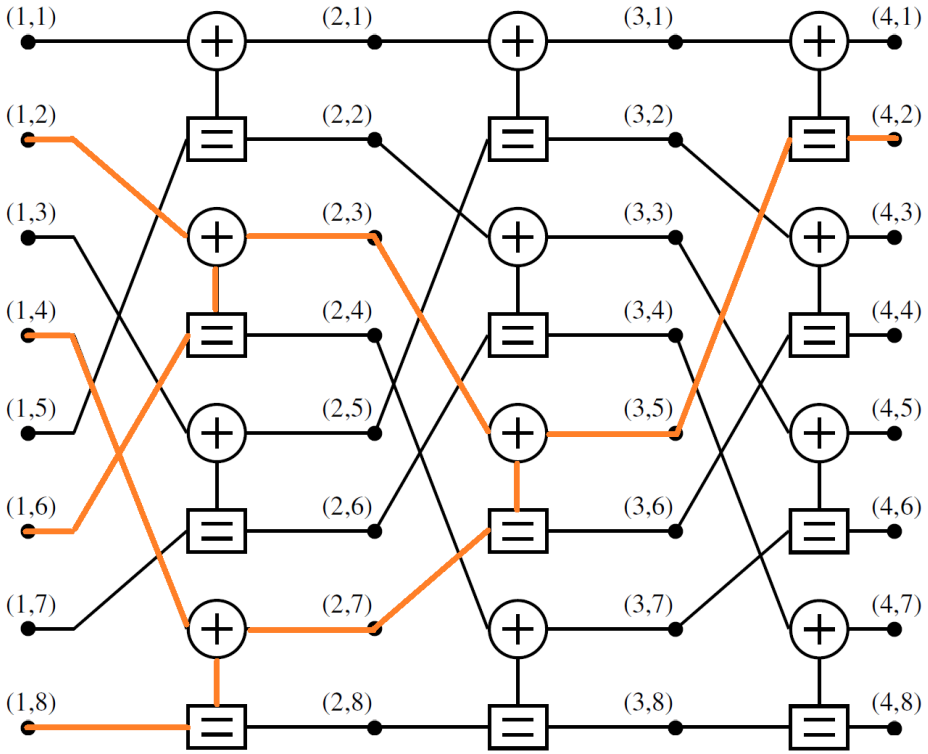




Uniform Factor Graph Representation

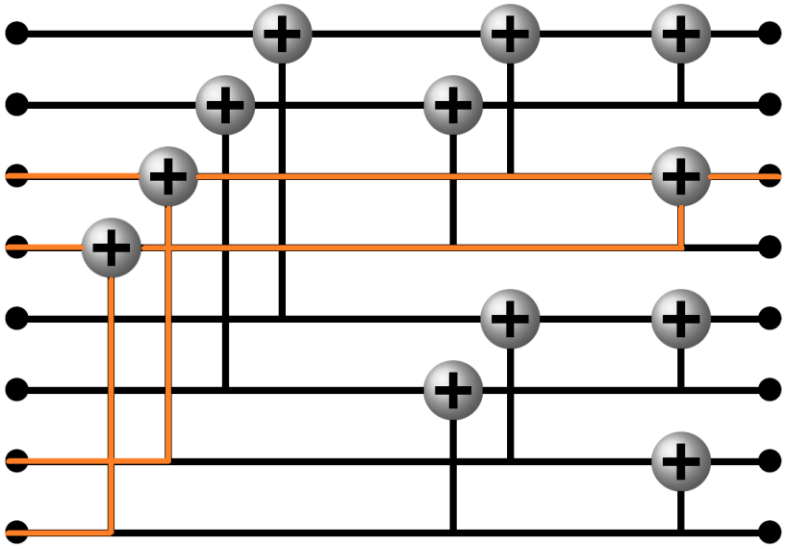
- The following two graphs do exactly the same thing.
- Remind that u_1^8 should be operated by bit-reversal before input to the graph to complete the encoding.

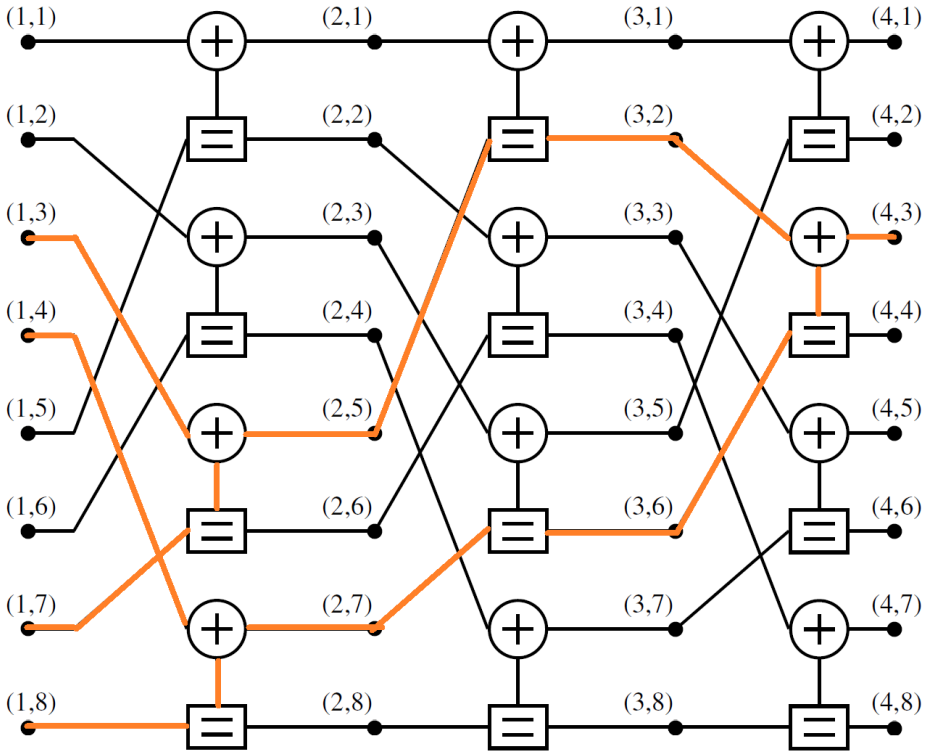




Uniform Factor Graph Representation

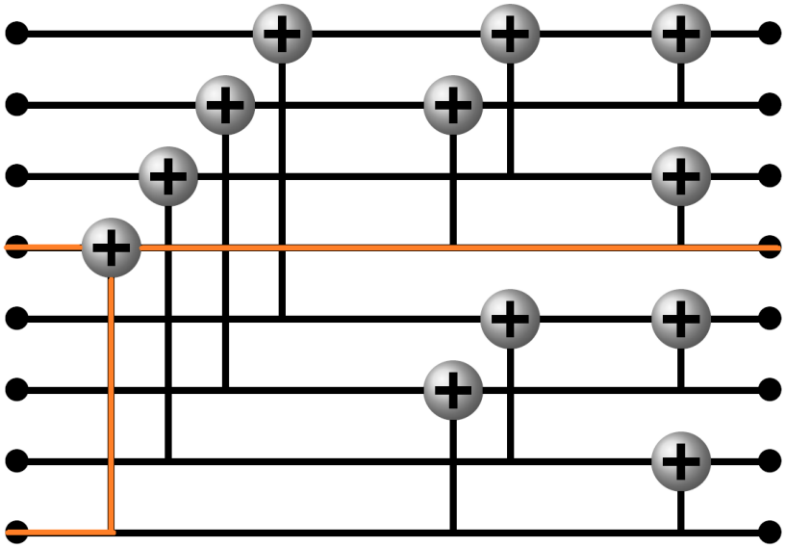
- The following two graphs do exactly the same thing.
- Remind that u_1^8 should be operated by bit-reversal before input to the graph to complete the encoding.

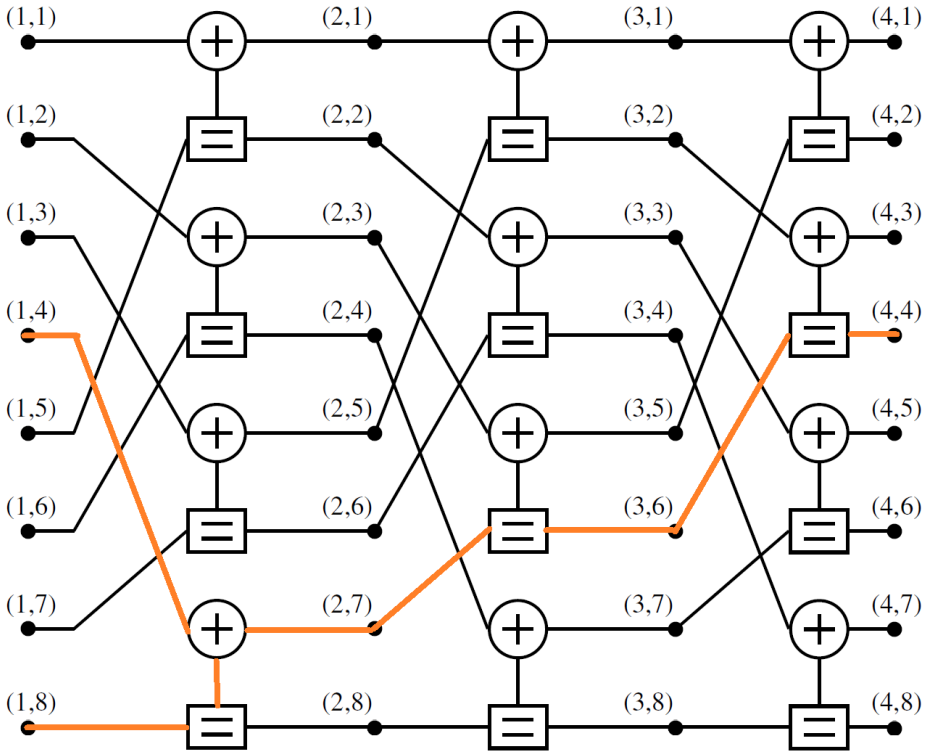




Uniform Factor Graph Representation

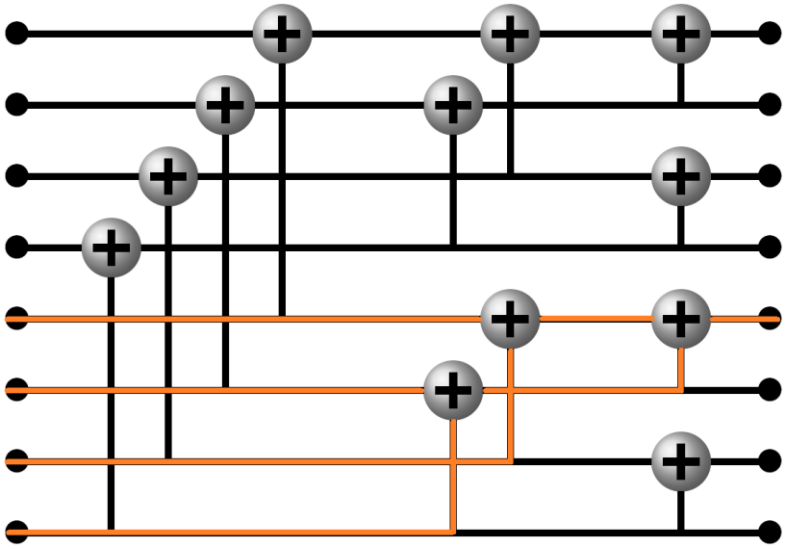
- The following two graphs do exactly the same thing.
- Remind that u_1^8 should be operated by bit-reversal before input to the graph to complete the encoding.

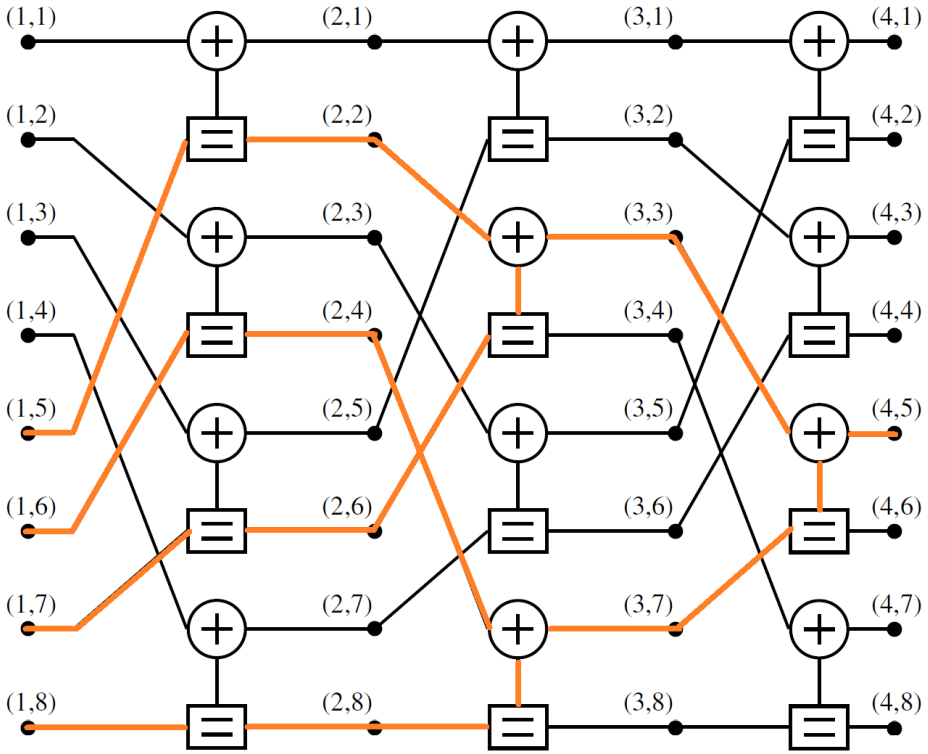




Uniform Factor Graph Representation

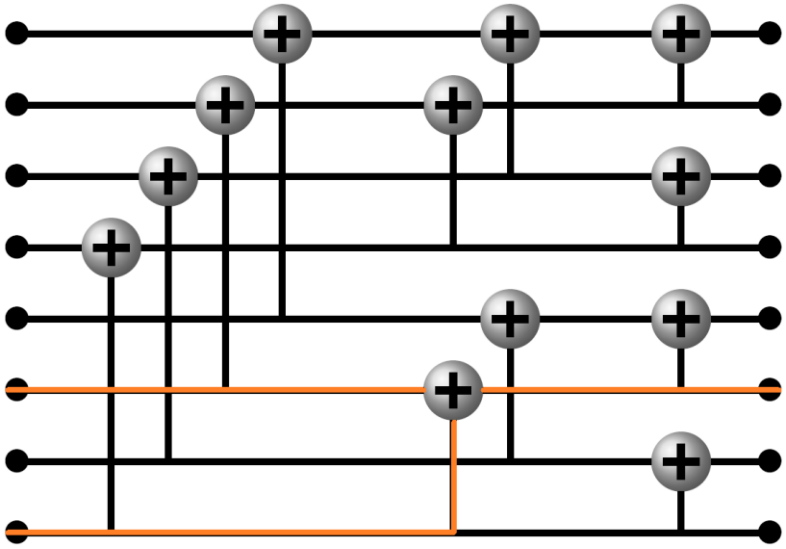
- The following two graphs do exactly the same thing.
- Remind that u_1^8 should be operated by bit-reversal before input to the graph to complete the encoding.

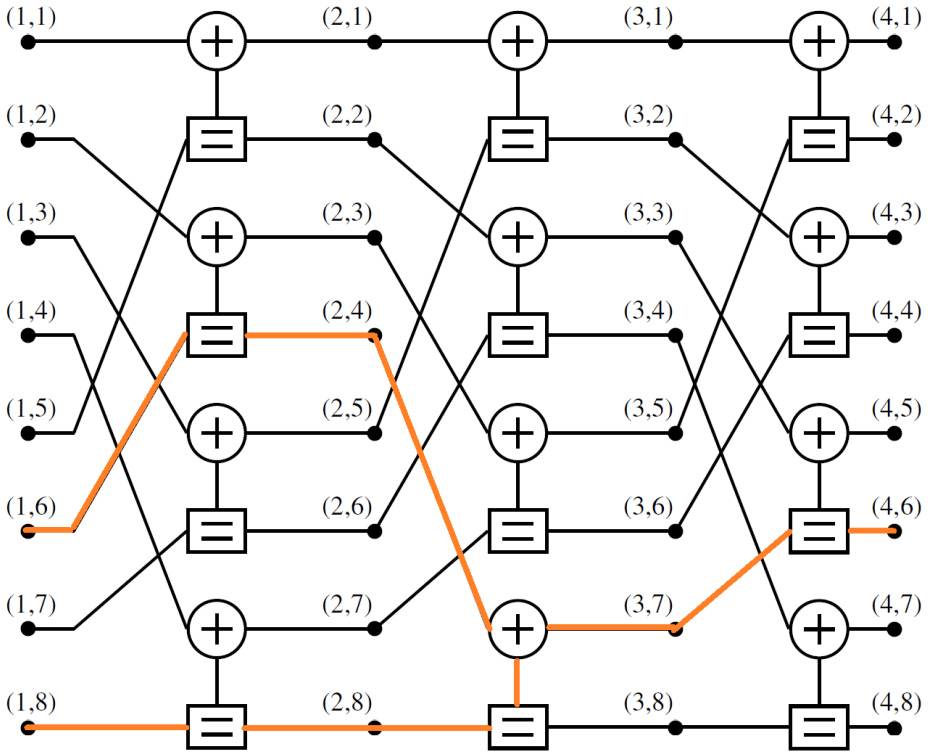




Uniform Factor Graph Representation

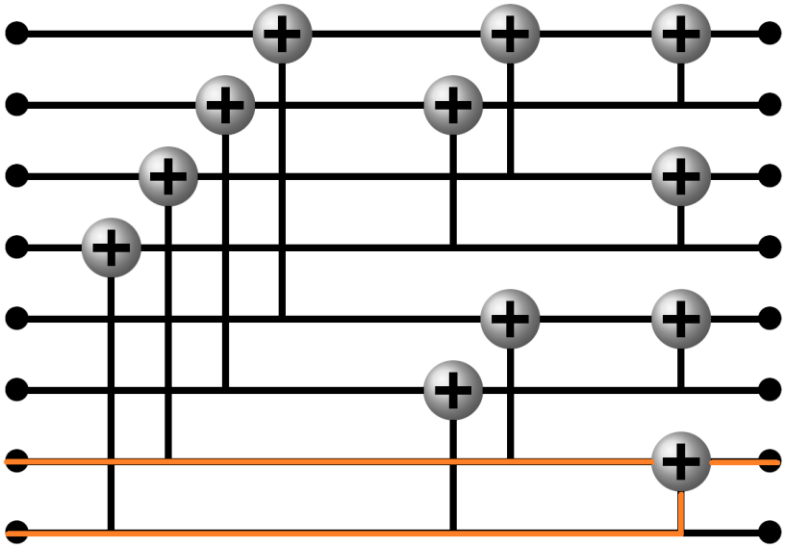
- The following two graphs do exactly the same thing.
- Remind that u_1^8 should be operated by bit-reversal before input to the graph to complete the encoding.

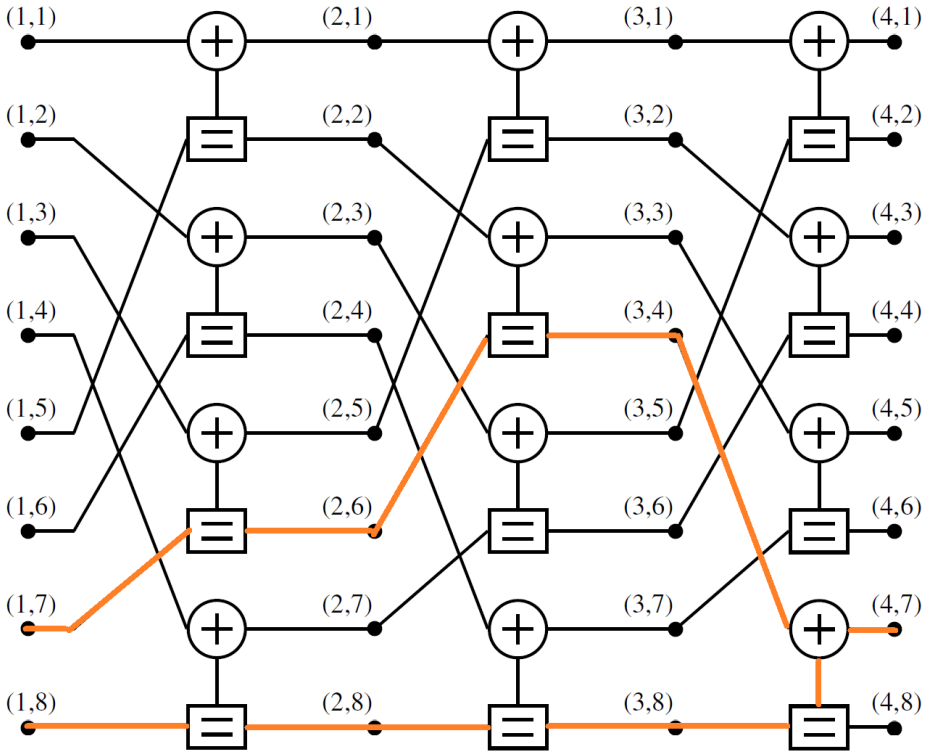




Uniform Factor Graph Representation

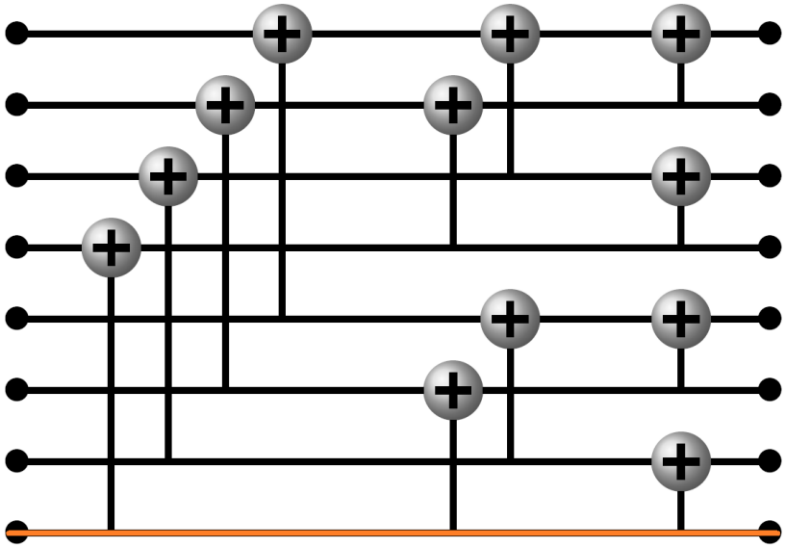
- The following two graphs do exactly the same thing.
- Remind that u_1^8 should be operated by bit-reversal before input to the graph to complete the encoding.

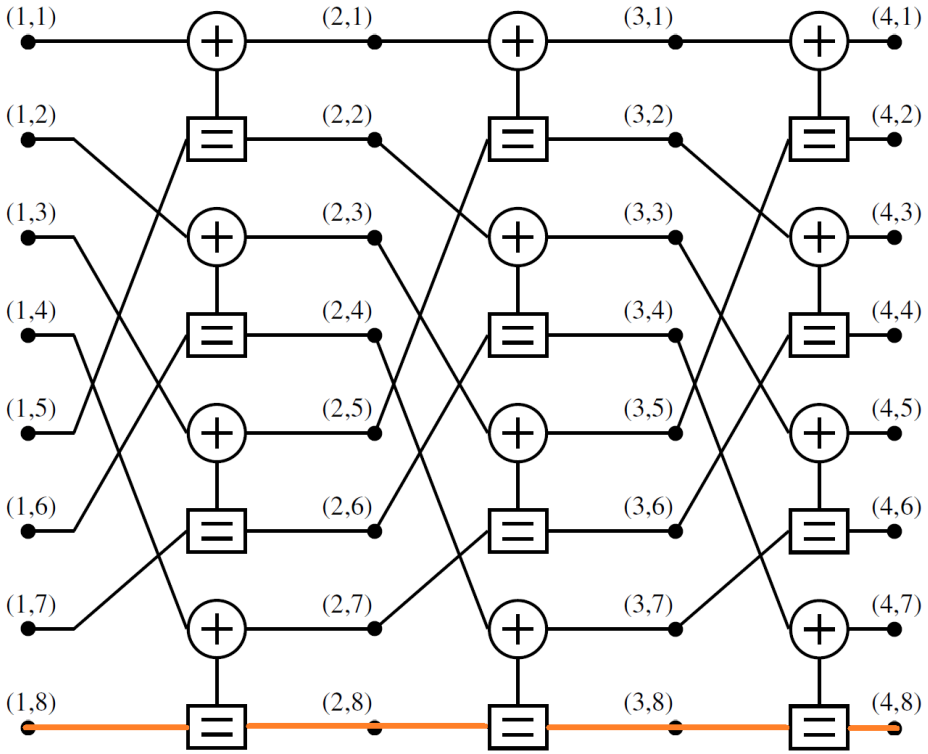




Uniform Factor Graph Representation

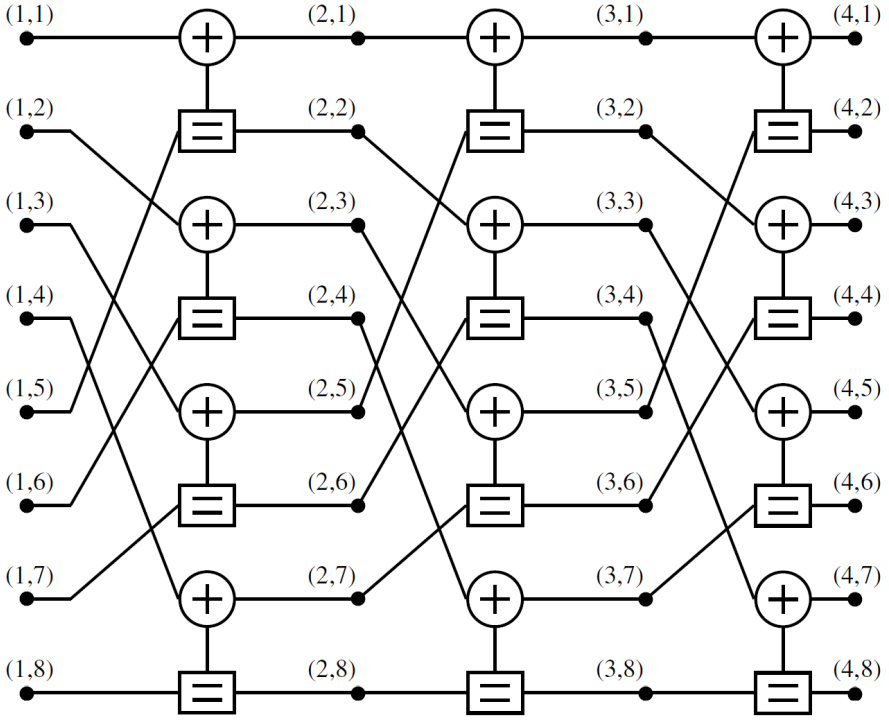
- The following two graphs do exactly the same thing.
- Remind that u_1^8 should be operated by bit-reversal before input to the graph to complete the encoding.





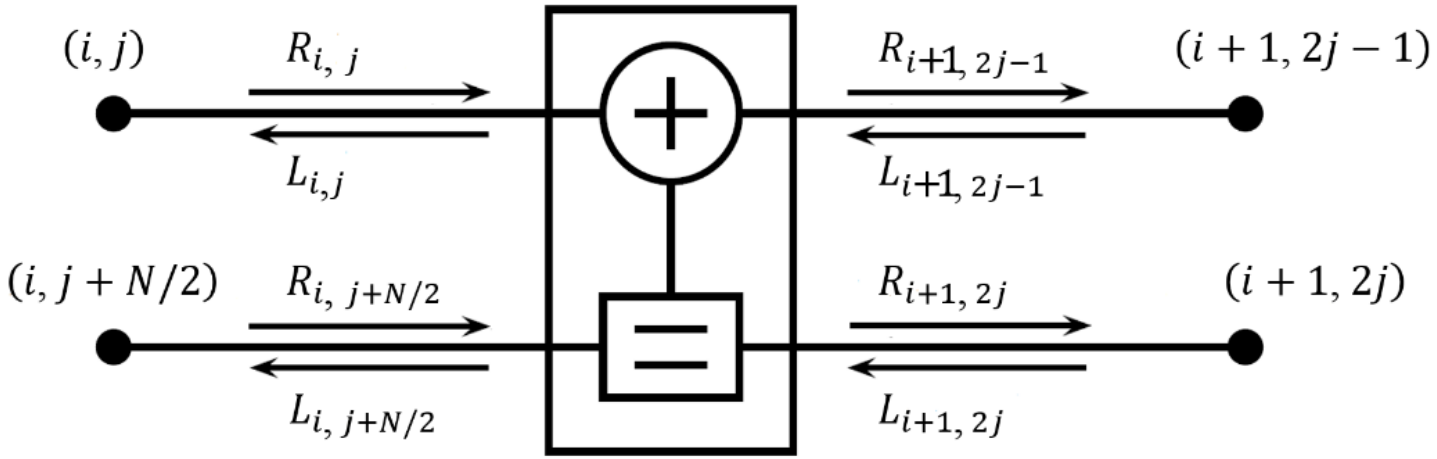
Uniform Factor Graph Representation

- Nodes $(1, *)$ are associated with the source vector $\mathbf{u}$.
- Nodes $(n+1, *)$ are associated with the codeword $\mathbf{x}$.
- For a node (i,j)
  - i denotes the i-th level counted from left to right.
  - j denotes the j-th position counted from above to bottom.



Basic Computational Block

- Nodes in the factor graph appear in groups of four, as forming the ports of 2-by-2 basic computational blocks (BCB).
- There are 4 nodes in a BCB.

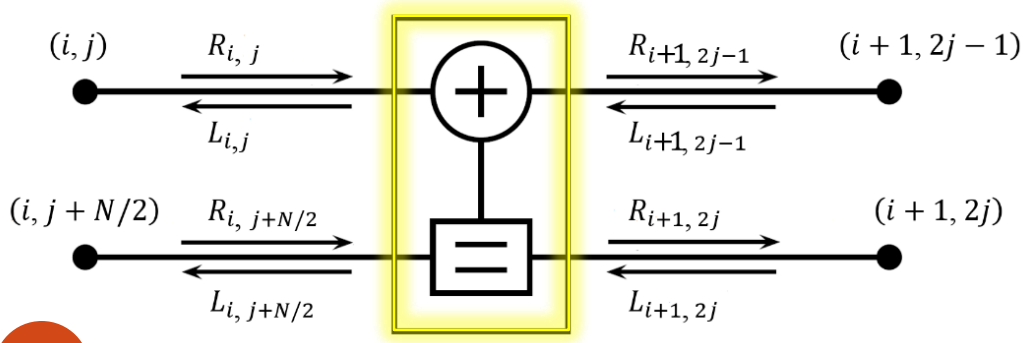


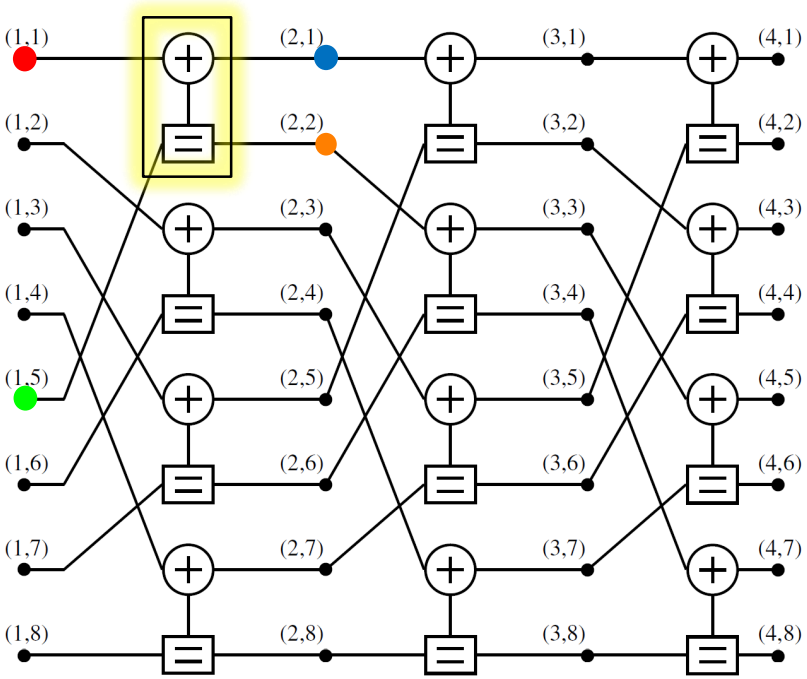
Basic Computational Block

- For a node (i,j)
  - i denotes the i-th level counted from left to right.
  - j denotes the j-th position counted from above to bottom.

- For $i = 1, j = 1$, nodes in a BCB are:

(1,1) (2,1)
(1,5) (2,2)



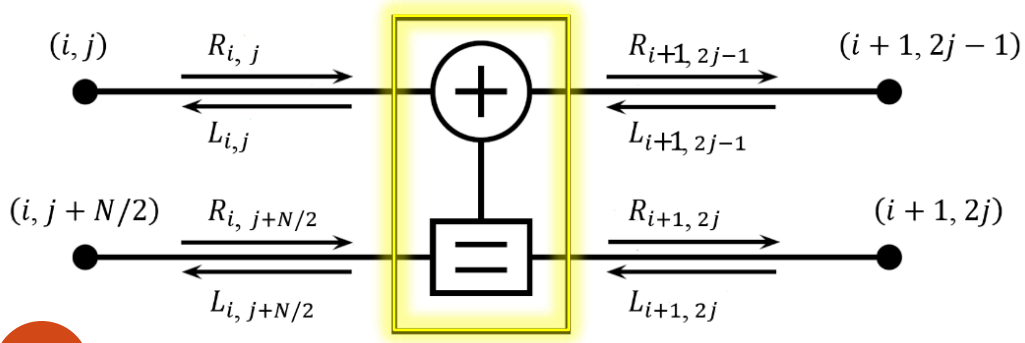


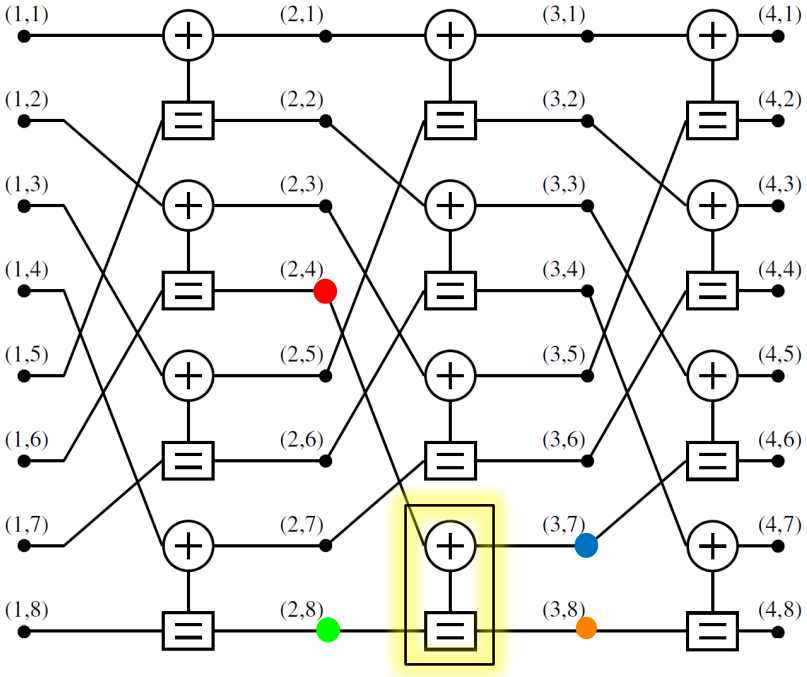
Basic Computational Block

- For a node (i,j)
  - i denotes the i-th level counted from left to right.
  - j denotes the j-th position counted from above to bottom.

- For $i = 2, j = 4$, nodes in a BCB are:

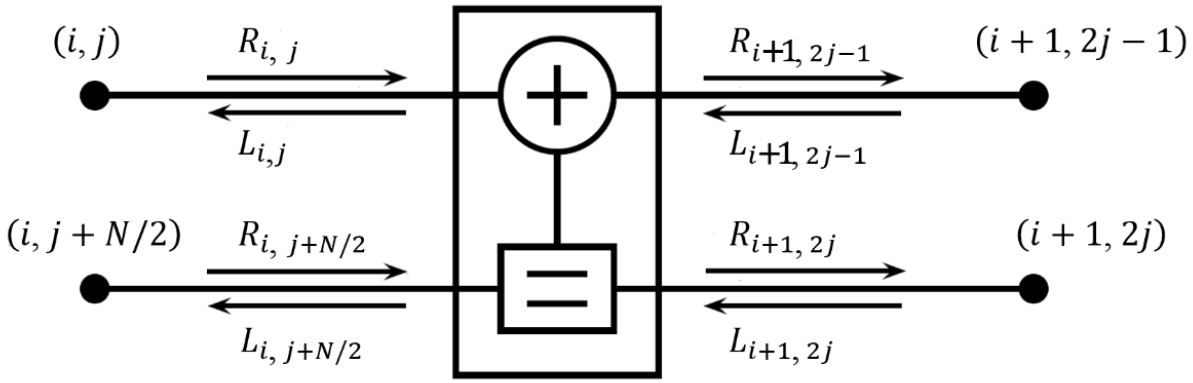
(2,4) (3,7)
(2,8) (3,8)





Basic Computational Block

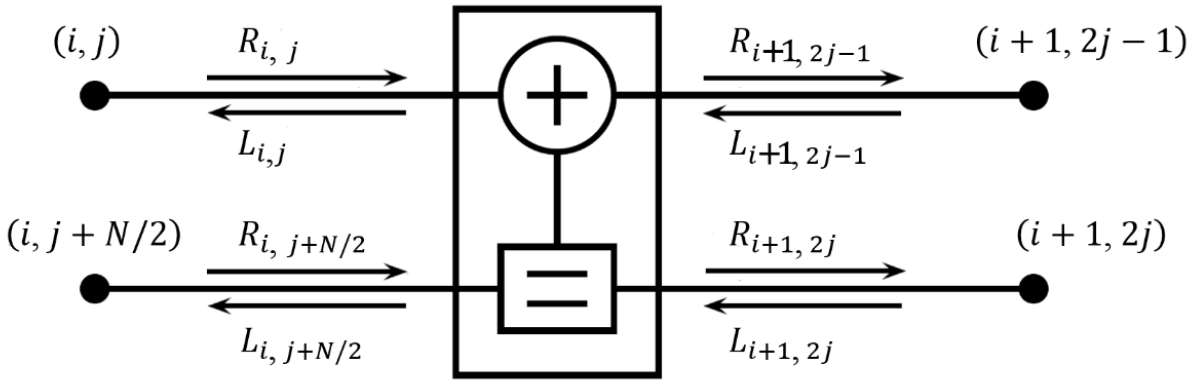
- The decoder is a message-passing decoder which computes the messages and the nodes simply relay the messages between neighboring basic computational blocks (BCB).
- The message that crosses node (i, j) from right to left (left to right) is designated by $L_{i,j}$ ($R_{i,j}$).



Log-Likelihood Ratios

- The messages $R_{1,j}$:

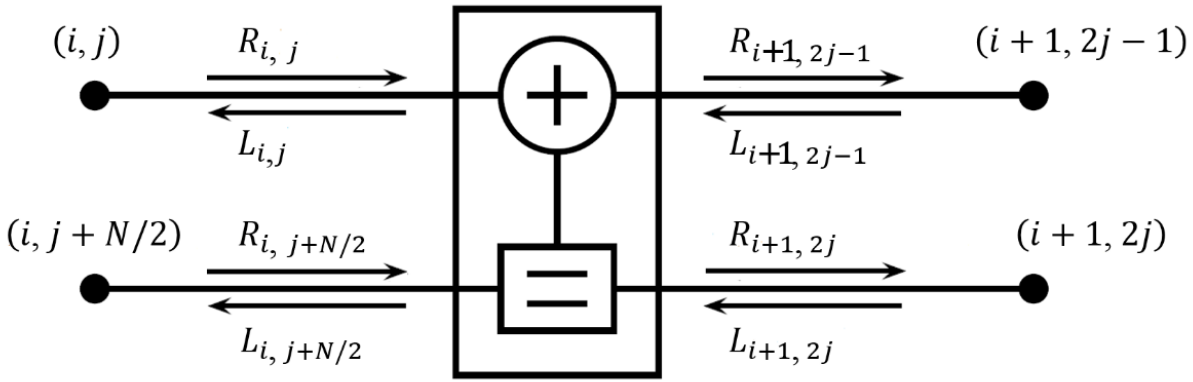
$$R_{1,j} = \begin{cases} 0 & \text{if } j \in \mathcal{A} \\ \infty & \text{if } j \in \mathcal{A}^c \text{ and } u_j = 0 \\ -\infty & \text{if } j \in \mathcal{A}^c \text{ and } u_j = 1 \end{cases}, 1 \le j \le N$$



Log-Likelihood Ratios

- The messages $L_{n+1,j}$:

$$L_{n+1,j} = \ln \frac{W(y_j|x_j=0)}{W(y_j|x_j=1)} \ , 1 \leq j \leq N$$



Log-Likelihood Ratios

- The messages-passing:

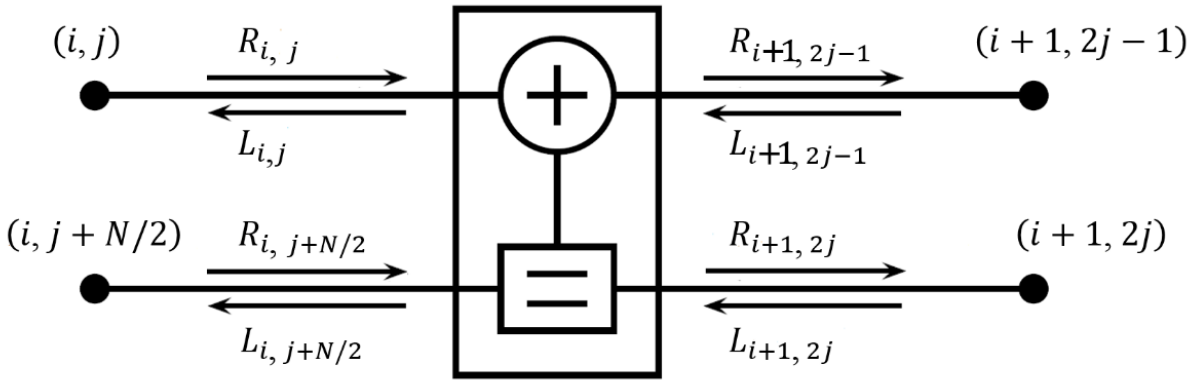
$$L_{i,j} = f(L_{i+1,2j-1}, L_{i+1,2j} + R_{i,j+N/2})$$

$$L_{i,j+N/2} = f(R_{i,j}, L_{i+1,2j-1}) + L_{i+1,2j}$$

$$R_{i+1,2j-1} = f(R_{i,j}, L_{i+1,2j} + R_{i,j+N/2})$$

$$R_{i+1,2j} = f(R_{i,j}, L_{i+1,\,2j-1}) + R_{i,j+N/2}$$

$$f(x,y) = \ln((xy+1)/(x+y)) \approx \mathrm{sgn}(x)\mathrm{sgn}(y)\min(|x|,|y|)$$



Log-Likelihood Ratios

- The messages-passing:

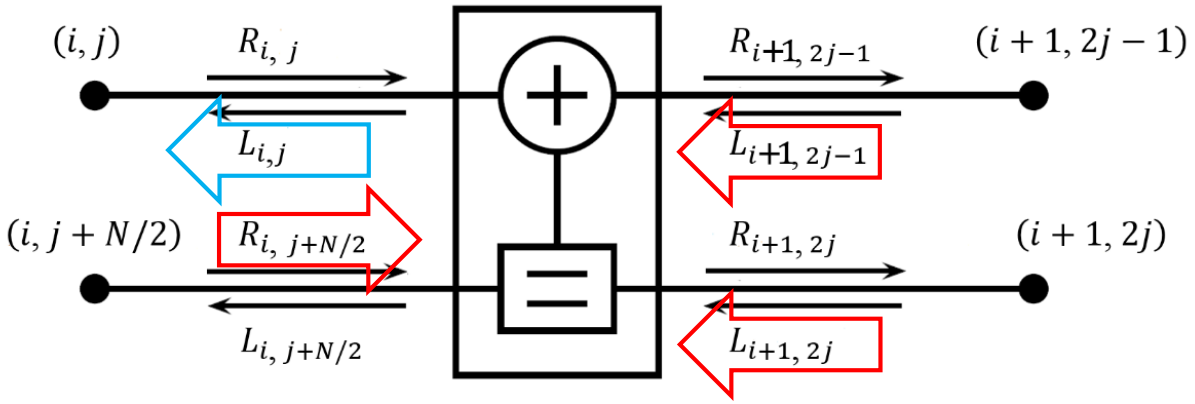
$$L_{i,j} = f(L_{i+1,2j-1}, L_{i+1,2j} + R_{i,j+N/2})$$

$$L_{i,j+N/2} = f(R_{i,j}, L_{i+1,2j-1}) + L_{i+1,2j}$$

$$R_{i+1,2j-1} = f(R_{i,j}, L_{i+1,2j} + R_{i,j+N/2})$$

$$R_{i+1,2j} = f(R_{i,j}, L_{i+1,\,2j-1}) + R_{i,j+N/2}$$

$$f(x,y) = \ln((xy+1)/(x+y)) \approx \mathrm{sgn}(x)\mathrm{sgn}(y)\min(|x|,|y|)$$



Log-Likelihood Ratios

- The messages-passing:

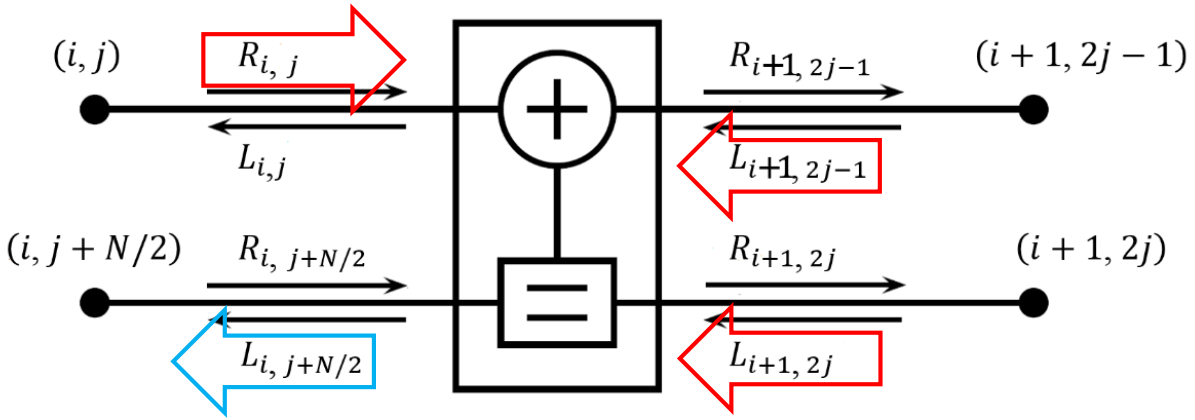
$$L_{i,j} = f(L_{i+1,2j-1}, L_{i+1,2j} + R_{i,j+N/2})$$

$$L_{i,j+N/2} = f(R_{i,j}, L_{i+1,2j-1}) + L_{i+1,2j}$$

$$R_{i+1,2j-1} = f(R_{i,j}, L_{i+1,2j} + R_{i,j+N/2})$$

$$R_{i+1,2j} = f(R_{i,j}, L_{i+1,\,2j-1}) + R_{i,j+N/2}$$

$$f(x,y) = \ln((xy+1)/(x+y)) \approx \mathrm{sgn}(x)\mathrm{sgn}(y)\min(|x|,|y|)$$



Log-Likelihood Ratios

- The messages-passing:

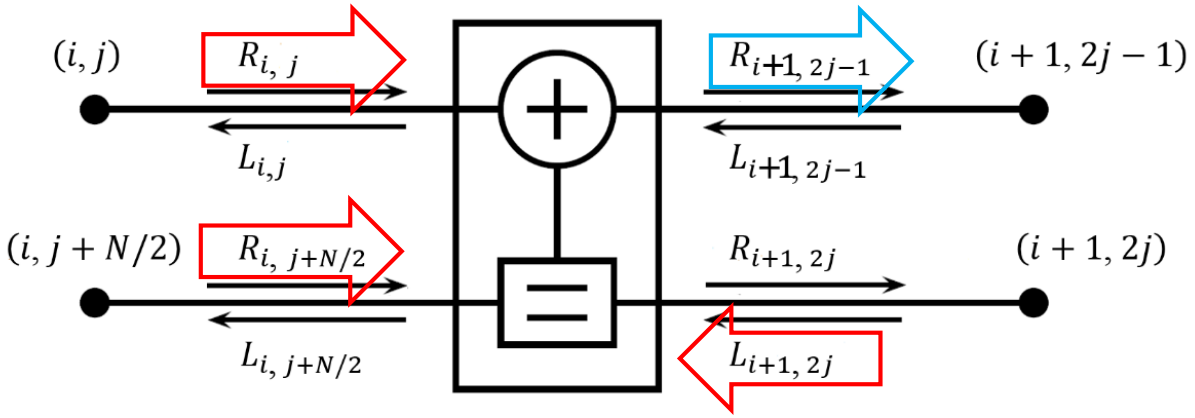
$$L_{i,j} = f(L_{i+1,2j-1}, L_{i+1,2j} + R_{i,j+N/2})$$

$$L_{i,j+N/2} = f\left(R_{i,j}, L_{i+1,2j-1}\right) + L_{i+1,2j}$$

$$R_{i+1,2j-1} = f(R_{i,j}, L_{i+1,2j} + R_{i,j+N/2})$$

$$R_{i+1,2j} = f\left(R_{i,j}, L_{i+1,\,2j-1}\right) + R_{i,j+N/2}$$

$$f(x,y) = \ln((xy+1)/(x+y)) \approx \mathrm{sgn}(x)\mathrm{sgn}(y)\min(|x|,|y|)$$



Log-Likelihood Ratios

- The messages-passing:

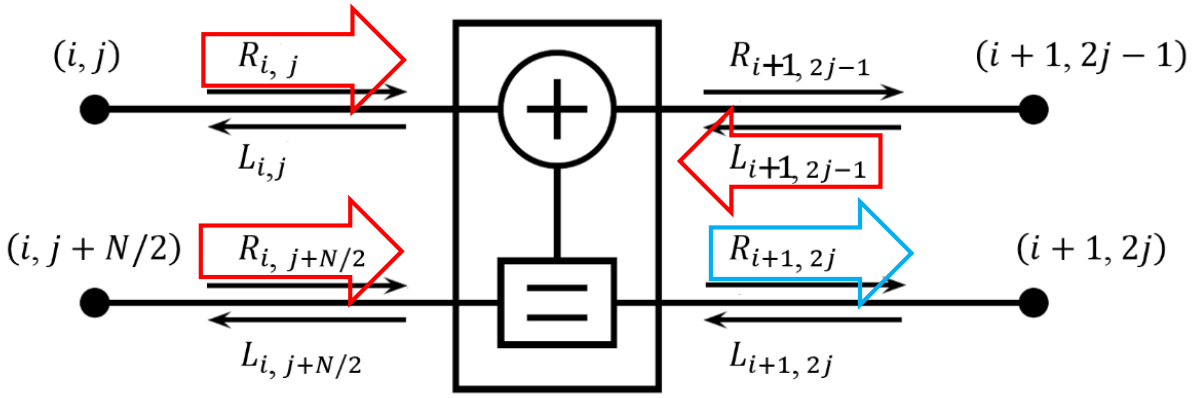
$$L_{i,j} = f(L_{i+1,2j-1}, L_{i+1,2j} + R_{i,j+N/2})$$

$$L_{i,j+N/2} = f(R_{i,j}, L_{i+1,2j-1}) + L_{i+1,2j}$$

$$R_{i+1,2j-1} = f(R_{i,j}, L_{i+1,2j} + R_{i,j+N/2})$$

$$R_{i+1,2j} = f(R_{i,j}, L_{i+1,\,2j-1}) + R_{i,j+N/2}$$

$$f(x,y) = \ln((xy+1)/(x+y)) \approx \mathrm{sgn}(x)\mathrm{sgn}(y)\min(|x|,|y|)$$



Summary

- Polar codes are proved to achieve the symmetric capacity of the binary-input discrete memoryless channels (B-DMCs).
- With the property of channel polarization, information bits are placed in the “good channels”, and the rest of channels are called frozen bits.
- Although polar codes have astonishing asymptotic performance, the finite-length performance of polar codes under SC decoding is unsatisfying.

Summary

- To further improve the performance of polar codes, many decoding algorithms have been proposed. There are two types of main decoding algorithms of polar codes, one is SC decoding and the other one is BP decoding.
- SC decoding exploits the serial processing scheme to decode by using the active level trees.

Summary

- LSC decoding allows more than one partial sequence to be further processed during the decoding process compared to the original SC decoding.
- CRC-aided LSC decoding using the concatenated CRC check to further improve the performance.
- BP decoding exploits the message passing method. With the parallel processing, BP decoding has the properties of lower latency and higher throughput compared to SC decoding.

References

- E. Arıkan, “Channel polarization: a method for constructing capacity achieving codes for symmetric binary-input memoryless channels,” *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.
- K. Chen, K. Niu, and J. R. Lin, “List successive cancellation decoding of polar codes,” *Electron. Lett.*, vol. 48, no. 9, pp. 500–501, 2012.
- K. Niu and K. Chen, “CRC-aided decoding of polar codes,” *IEEE Communications Letters*, vol. 16, no. 10, October 2012.
- A. Pamuk, “An FPGA implementation architecture for decoding of polar codes,” in Proc. *18th ISWCS*, 2011, pp. 437–441.
- Gabi Sarkis, Pascal Giard, Alexander Vardy, Claude Thibeault, and Warren J. Gross, “Fast List Decoders for Polar Codes ,” *IEEE Journal on Selected Areas in Communications*, Vol. 34, no. 2, pp.318–328, Feb 2016.
- I. Tal and A. Vardy, “List decoding of polar codes,” *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.
- http://www.polarcodes.com/