

ALTERA.
UNIVERSITY
PROGRAM

5G

The 1st 5G Algorithm
Innovation Competition
Polar Code

Organizer |



西安电子科技大学
XIDIAN UNIVERSITY

terasic Terasic Inc.

Sponsor |



SPREADTRUM

InnovateAsia-1st 5G Mobile Algorithm Competition-Polar Code

Task	Description	Requirements
<p>Polar Code - An capacity-achieving error correcting code</p>	<p>Channel coding technique ensures reliable and efficient communications in modern systems;</p> <p>Given a specific channel, the highest code rate for reliable communications is limited by Shannon's channel capacity;</p> <p>Some good existing channel codes, which include turbo codes and LDPC codes, can approach (but achieve) the channel capacity;</p> <p>Polar codes can achieve the channel capacity with a constructive structure and simple encoding and decoding schemes;</p> <p>Under practical and finite code length configuration, polar code can achieve a better error-correction performance than turbo/LDPC codes with comparable complexity.</p> <p>Based on the materials provided herein and the public references, understand the encoding and decoding procedures of the polar, develop simulation program and then generate correct simulation results. Design FPGA implementation architectures for polar encoder and decoder; implement the FPGA design using Verilog RTL hardware description language; conduct the FPGA test using the hardware and tools provided by the contest organizer, and then deliver the corresponding testing results.</p>	<p>Requirements:</p> <p>Implement software and hardware polar encoder and decoder based on the provided materials and other references on the Internet.</p> <p>Works format:</p> <ol style="list-style-type: none">1. Algorithm study: polar encoding and decoding algorithms, C/Matlab simulation;2. FPGA implementation: implement polar encoder and decoder using hardware and tools provided in this contest. <p>Delivery Material:</p> <ol style="list-style-type: none">1. Algorithm documents: algorithm description and simulation results, C/Matlab code files;2. FPGA design document, Verilog RTL hardware description program, FPGA downloading bit files and testing results. <p>Selection criteria in the first round</p> <ol style="list-style-type: none">1. Correctly understand the polar encoding and decoding algorithms; the software simulation results is reasonable;2. Good FPGA design for polar encoder and decoder. <p>Selection criteria in the second round</p> <ol style="list-style-type: none">1. Complete the FPGA implementation of polar encoder and decoder;2. High data throughput, low processing delay and less FPGA resources occupation is desired;3. Novel algorithm and architecture is a plus.

References to Read

MUST Read Papers:

- E. Arikan, "Channel polarization: **A method for constructing capacity achieving codes for symmetric binary-input memoryless channels**," IEEE Trans. Information Theory, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.
- A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "**LLR based successive cancellation list decoding of polar codes**," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.(ICASSP), May 2014, pp. 3903-3907.

Polar code Related Publications:

- B. Li, H. Shen, D. Tse, "**An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check** ", IEEE Communications Letters, Volume: 16 , Issue: 12, 2012 , Page(s): 2044 – 2047
- B. Li, H. Shen, D. Tse, "**Parallel decoders of polar codes**", arXiv:1309.1026, Sep., 2013.[Online]. Available: [http:// arxiv.org/pdf/1309.1026](http://arxiv.org/pdf/1309.1026)
- Y. Z. Fan, J. Chen, C. Xia, C. Y. Tsui, J. Jin, H. Shen, B. Li, "**Low latency list decoding of polar codes with double thresholding**," ICASSP, 2015
- I. Tal and A. Vardy, "**List decoding of polar codes**," arXiv:1206.0050v1, 2012 [Online]. Available: <http://arxiv.org/pdf/1206.0050v1>
- K. Niu, K. Chen, J. Lin, and Q. T. Zhang, "**Polar Codes: Primary Concepts and Practical Decoding Algorithms**", IEEE Communication Magazine, Volume: 52 , Issue: 7, 2014 , Page(s): 192 - 203

Outline

- **What is Polar Code?**
- **Why we need Polar Code?**
- **How does Polar Code Work?**
- **Implement Guide**

To have a gut feeling what is polar code

WHAT IS POLAR CODE?

What is Polar Code?



100101.....0
K-bit message

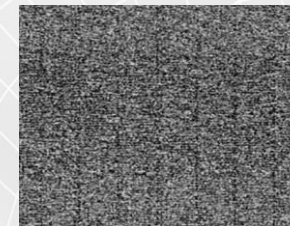


Transmitter

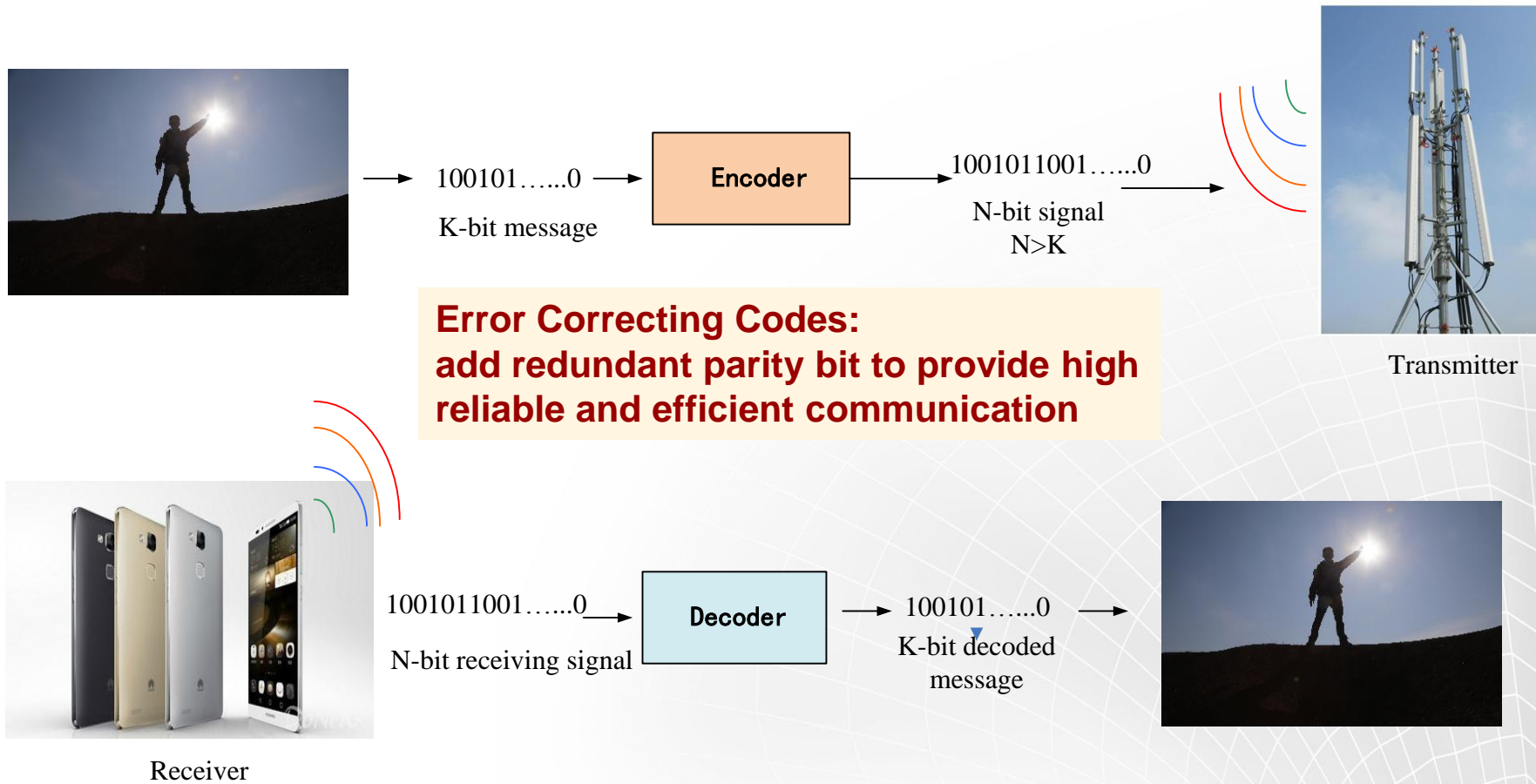


Receiver

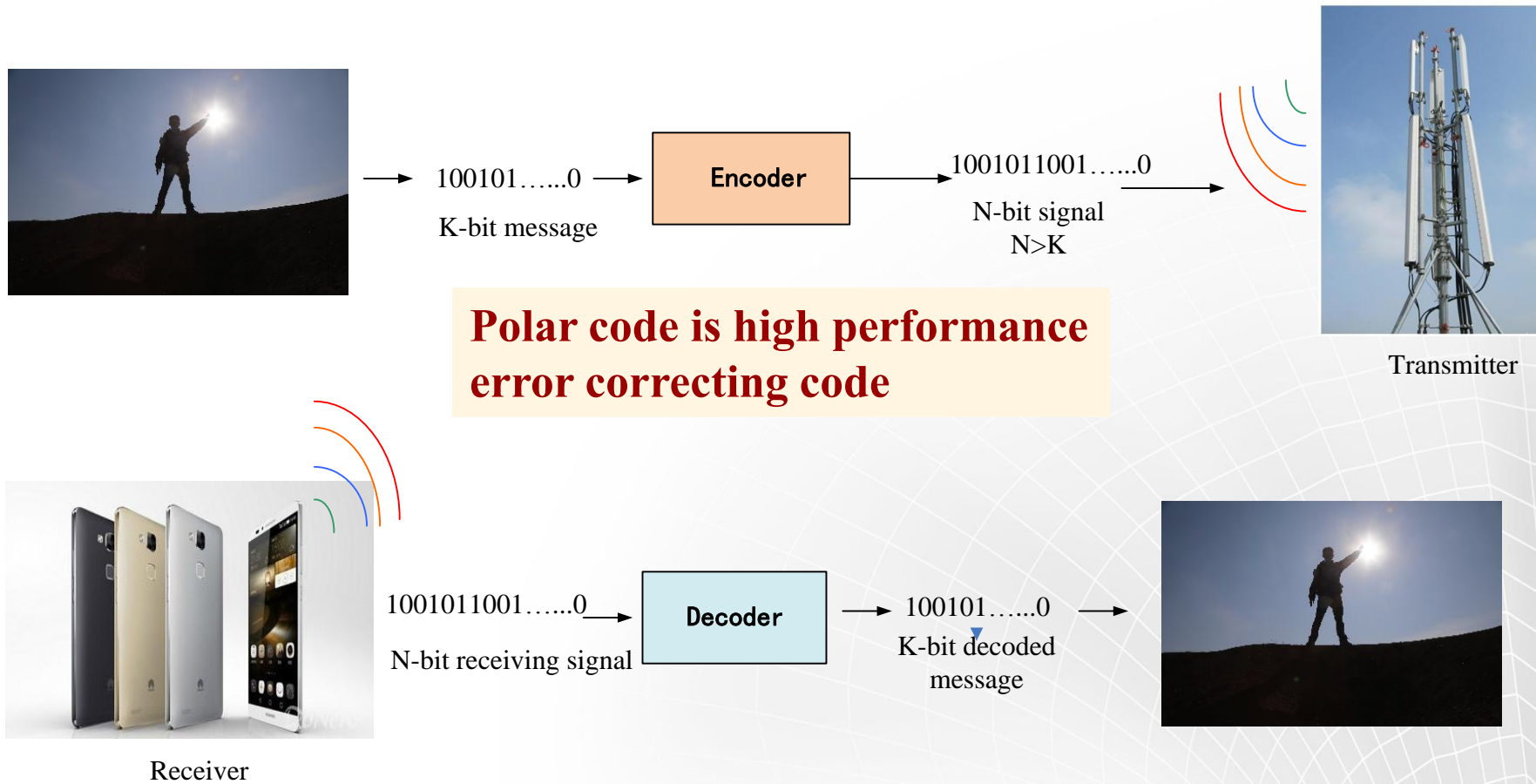
10?10?.....?
K-bit receiving signal



What is Polar Code?



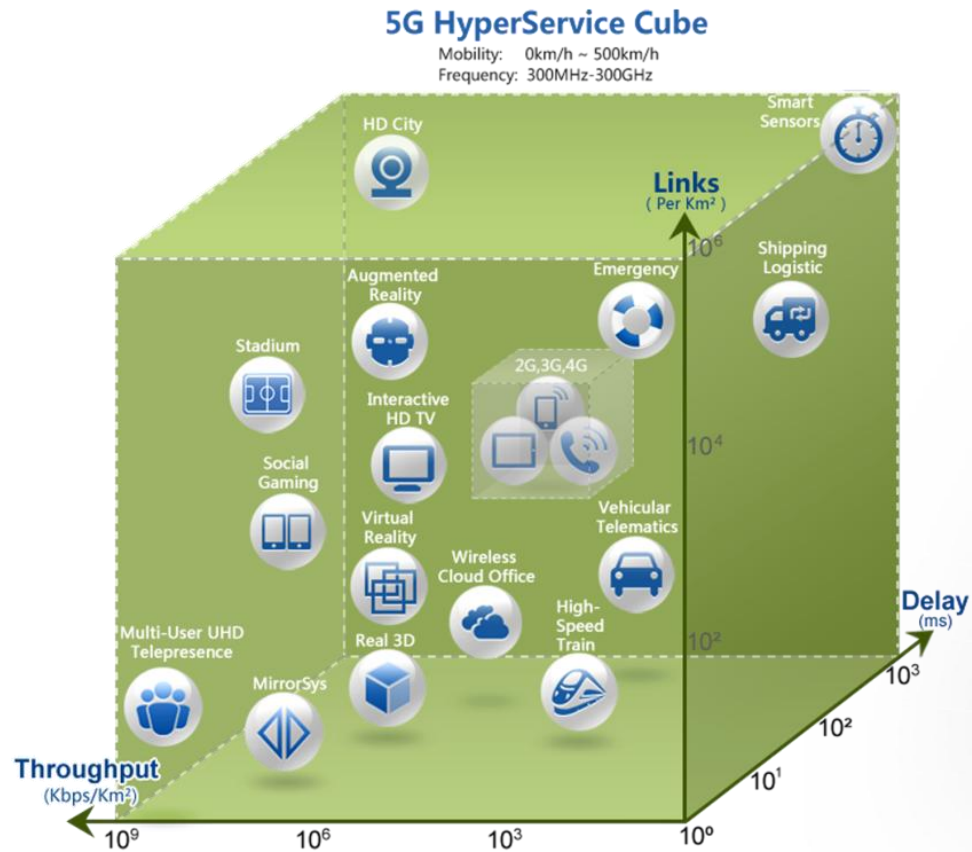
What is Polar Code?



To know what benefit polar code brings to 5G

WHY WE NEED POLAR CODE?

5G Vision: Zero Distance Communications



Massive Capacity



Massive Connectivity



Zero Waiting



10³
 Times Traffic Flooding

10²
 Billion Connections

10⁰
 ms Delay Experience

5G Requirements on Error Correcting Codes

■ Human Centric Communications: **HIGH PERFORMANCE, HIGH SPEED**

➤ User data rate: **10Gbps**

- iPhone, iPad, iGlass, iWatch

➤ Base station data rate: **1Tbps**

- Cloud computing blade

■ Machine Centric Communications: **LOW POWER, LOW LATENCY**

➤ Remote sensors **10~100Bytes**

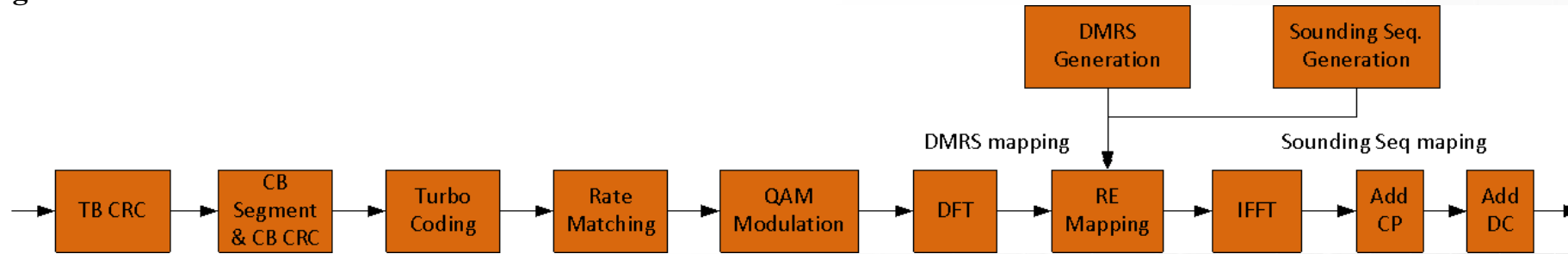
- meters, telemetric, RFID,

➤ Automation **10⁻⁴ second latency**

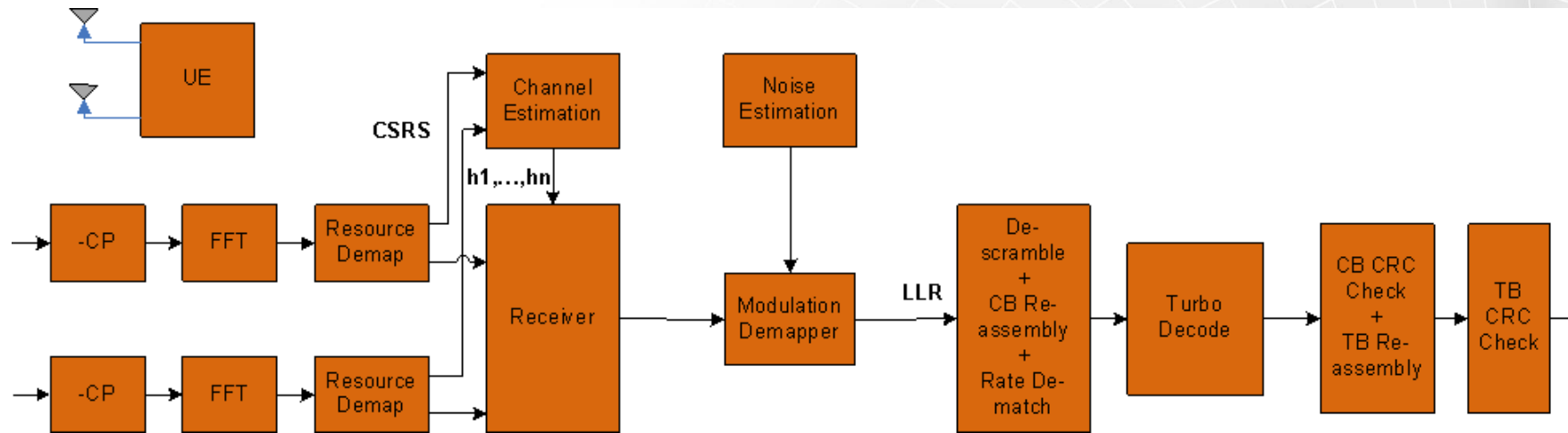
- Could-drive-car, factory control

Polar Code in Wireless System

Block diagram of base-band transmitter

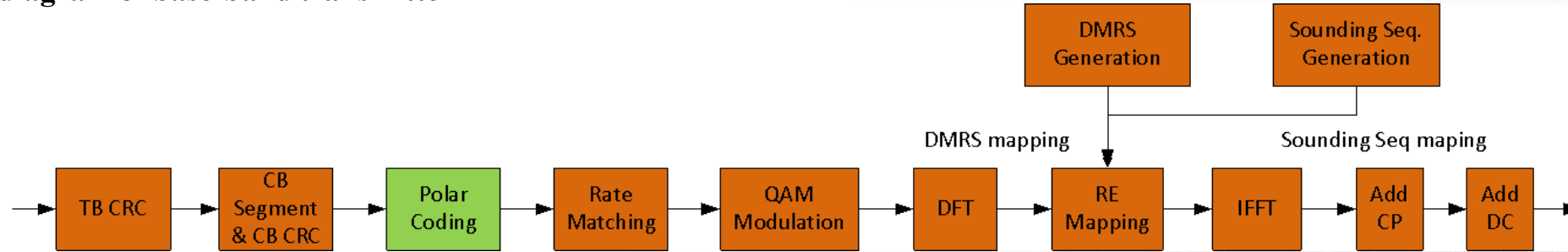


Block diagram of base-band receiver

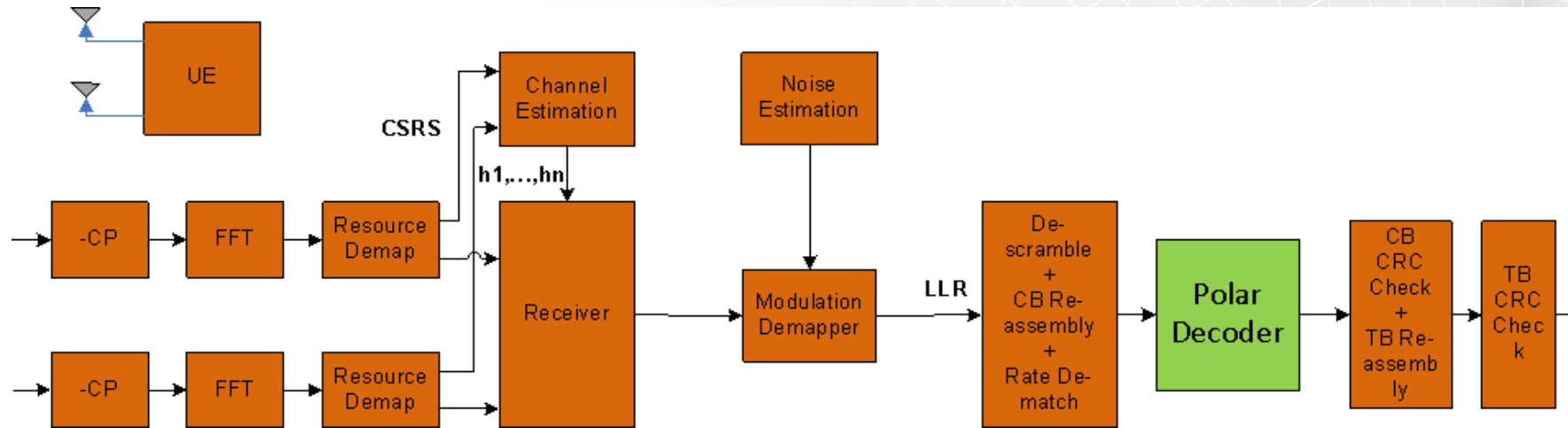


Polar Code in Wireless System

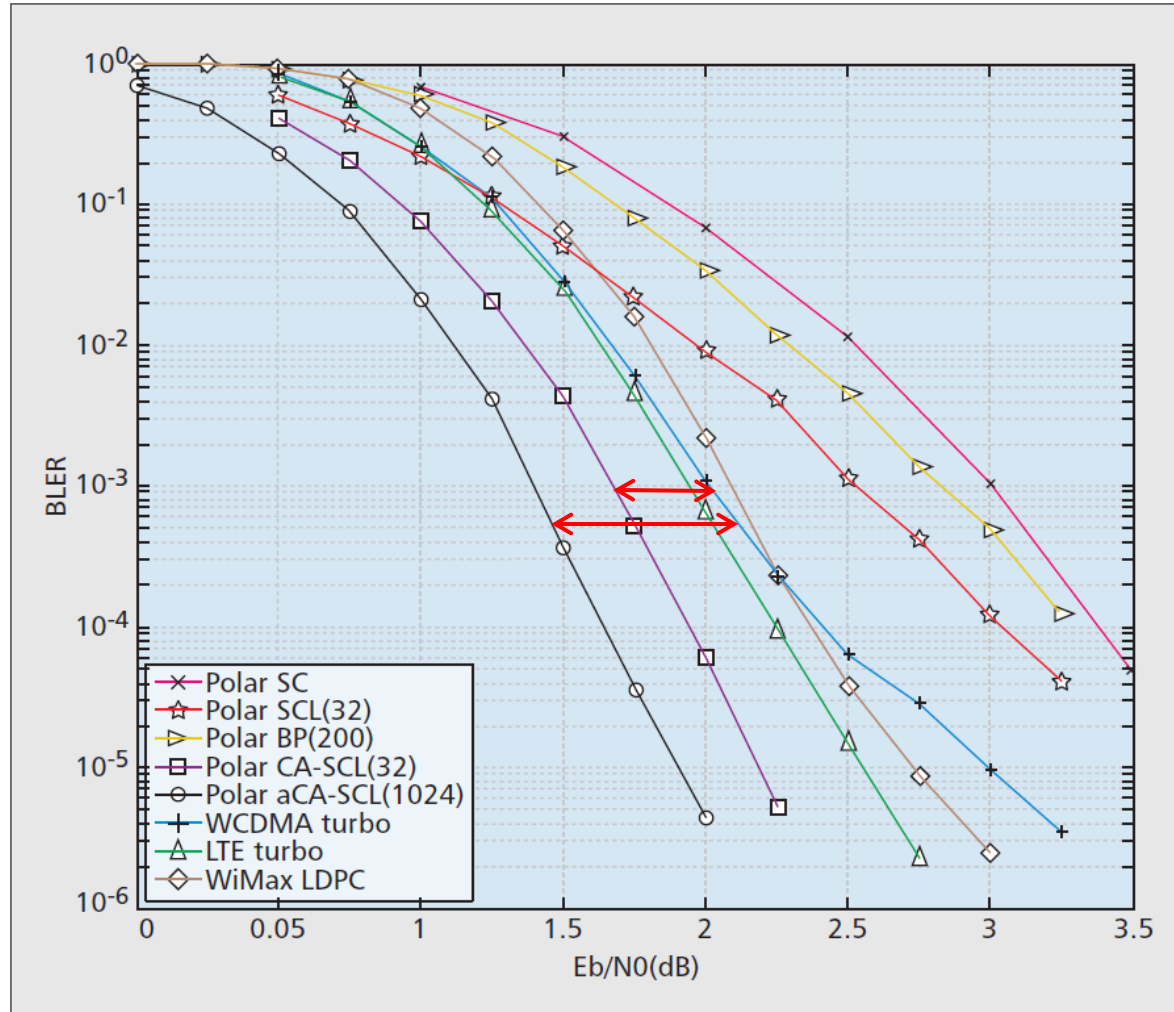
Block diagram of base-band transmitter



Block diagram of base-band receiver



Polar Code vs Classical Code



- Code: code length is 1024, code rate 0.5
- Channel: BPSK+AWGN
- Compared with Turbo/LDPC codes, polar code has 0.3~0.7dB performance improvement

To have a gut feeling how polar code will be implemented in the 5G wireless systems

HOW DOES POLAR CODE WORK?

Polar Encoding

- Combining the information and frozen bits $u_1^N = (u_1, u_2, \dots, u_N)$
 - K-length information bit sequence;
 - (N-K)-length frozen bit sequence, known by both encoder and decoder;
 - Combine two sequences, where u_A is the information bits and u_{A^c} is the frozen bits, the indices are collected in set A and its complement set A^c .

- Multiplied by the generator matrix

$$v_1^N = u_1^N F^{\otimes n}$$

where, $n = \log_2 N$, $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, $F^{\otimes n} = F \otimes F^{\otimes(n-1)}$, $\otimes n$ is the n-th Kronecker power.

- Perform bit-reversal permutation on v_1^N , i.e., $x_i = v_{\pi(i)}$
Function $\pi(i)$ is defined as follows: let the binary representation

of index i to be (b_1, b_2, \dots, b_n) , i.e., $i = \sum_{k=1}^n (b_k \cdot 2^{n-k}) + 1$

then, the binary representation of $\pi(i)$ is $(b_n, b_{n-1}, \dots, b_1)$.

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

$$F^{\otimes 2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$F^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Polar Encoding

Example: Polar Encoding with $N=8$ $K=4$

Information bit indices $A = \{4, 6, 7, 8\}$

Frozen bit indices $A^c = \{1, 2, 3, 5\}$

Information bit sequence (i_1, i_2, i_3, i_4)

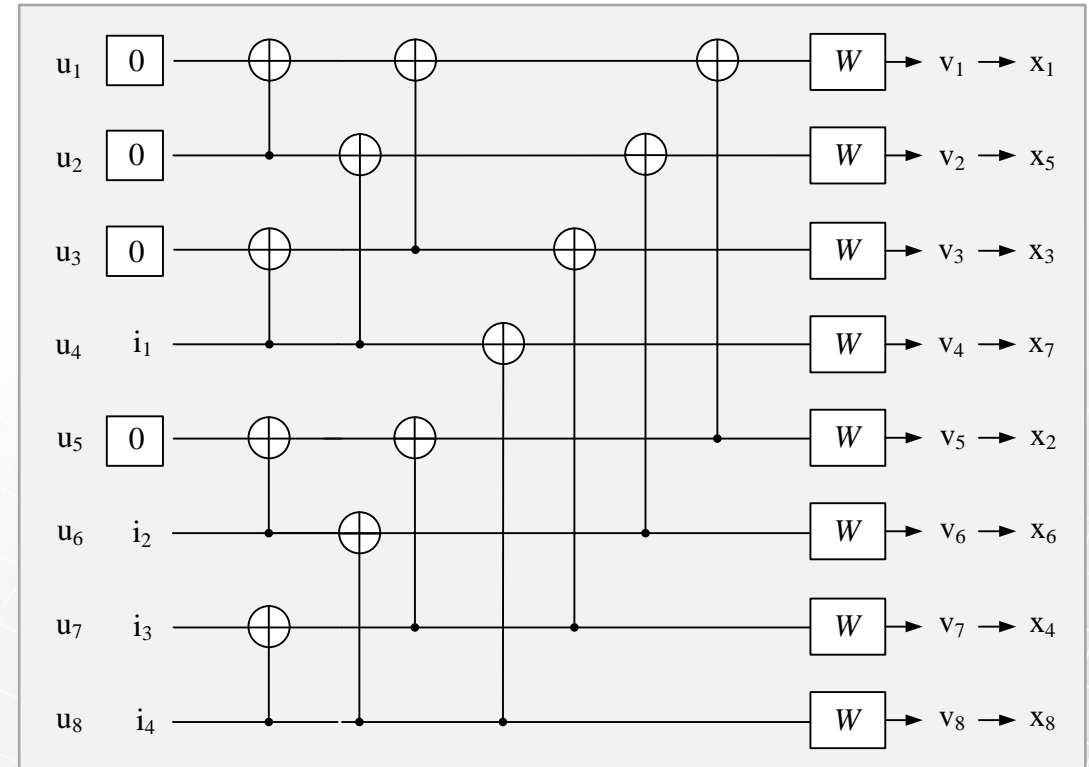
Frozen bit sequence $(0, 0, 0, 0)$

After bit combining, $u_1^8 = (0, 0, 0, i_1, 0, i_2, i_3, i_4)$

Multiplication $v_1^8 = u_1^8 F^{\otimes 3}$ can be realized using structure shown right

Bit-reversal indices of $(1, 2, 3, 4, 5, 6, 7, 8)$ is $(1, 5, 3, 7, 2, 6, 4, 8)$

Perform bit-reversal permutation on v_1^8 , and obtain the encoding bits x_1^8



Decoding Algorithm for Polar Codes

■ Successive cancellation (SC) decoding

- Low complexity, simple decoding structure;
- Be proven to achieve the channel capacity when the code length is large enough;
- Under practical finite length, performance is not satisfying.

■ Belief propagation and some other existing decoding algorithms are also not good enough

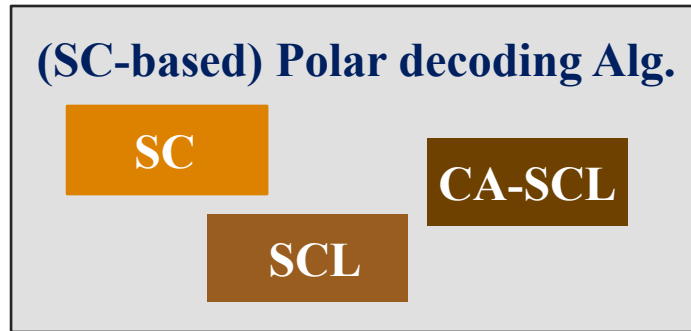
■ SC list (SCL) decoding algorithm

- Improve SC decoding algorithm;
- Achieve the ML performance with rather low complexity.

■ CRC-aided SCL (CA-SCL) decoding algorithm

- It's the normal case that CRC bits are embedded in the information sequence;
- Utilize the a priori information of CRC to make decision in the list obtained by SCL decoding;
- Achieve comparable and even better performance against Turbo/LDPC codes.

Decoding Algorithm for Polar Codes



Path search on the code tree



What is the TREE for polar decoding?

Definition of code tree

Which paths should be extended?

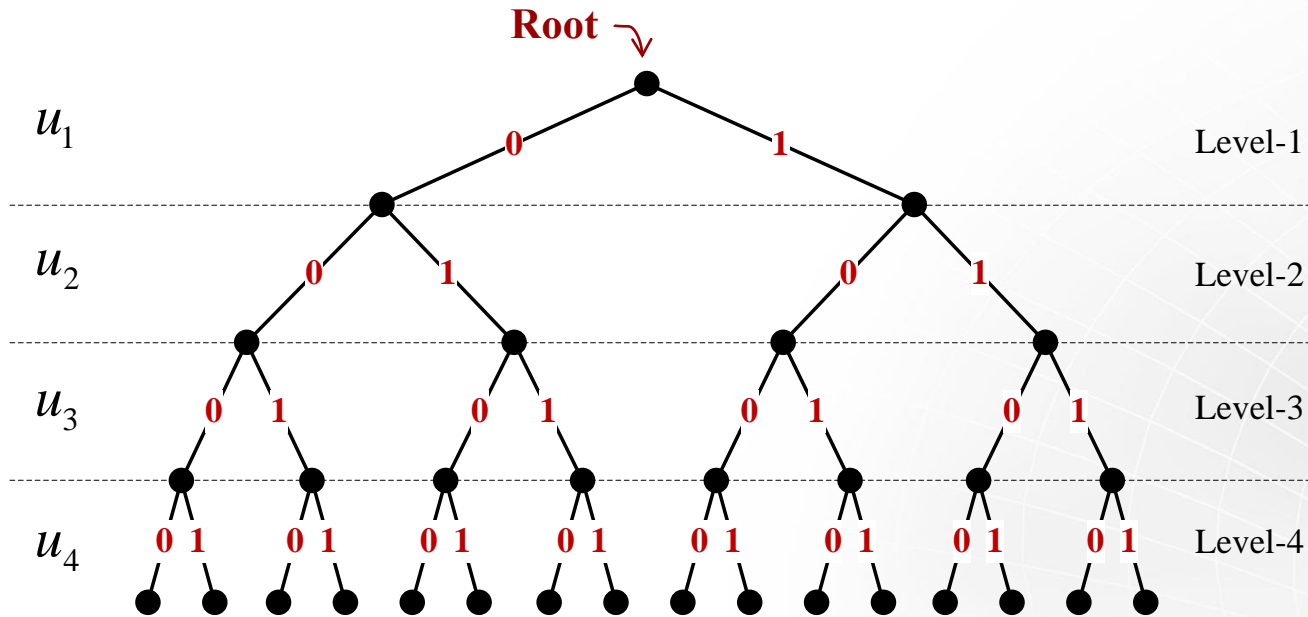
Rule of path expansion

How to calculate the path metrics?

Computation of path metric

Decoding Algorithm for Polar Codes

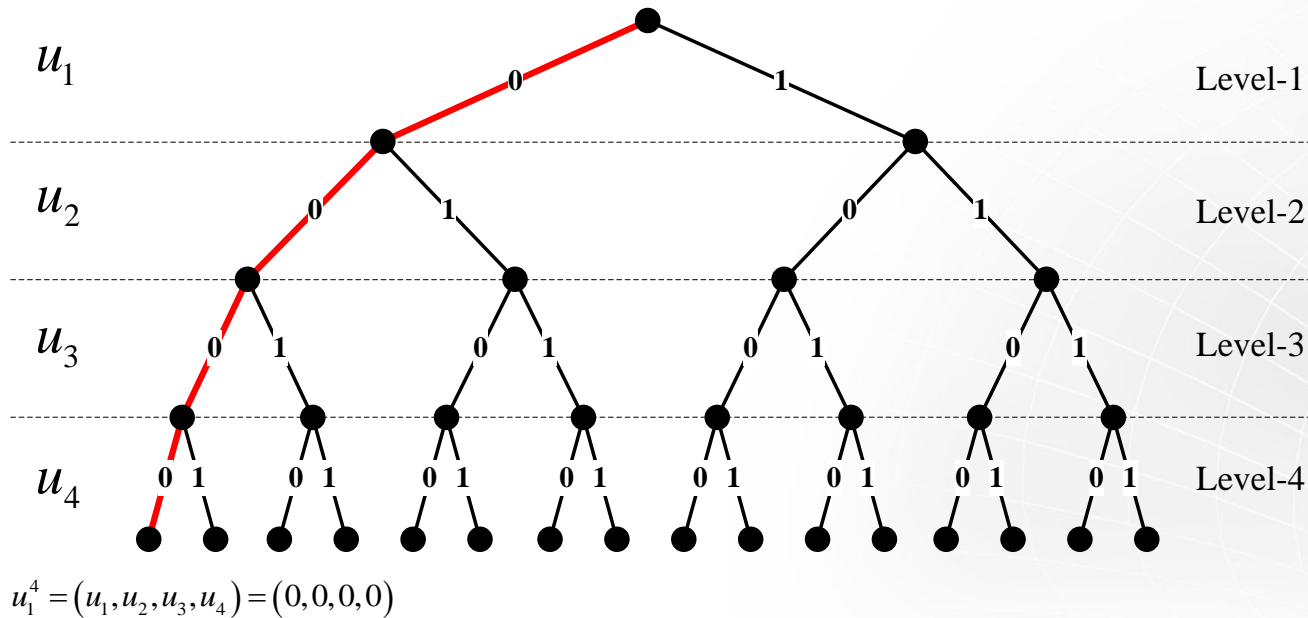
What is the TREE for polar decoding?



- Any polar code with length N is related to a full binary tree with depth N ;
- Each level of the edges is related to an information or a frozen bit;
- Except the leaf nodes, the two dependants of each node are respectively labeled as 0 and 1;
- Each path from the root to any leaf node is related to a decoding sequence (frozen bit included).

Decoding Algorithm for Polar Codes

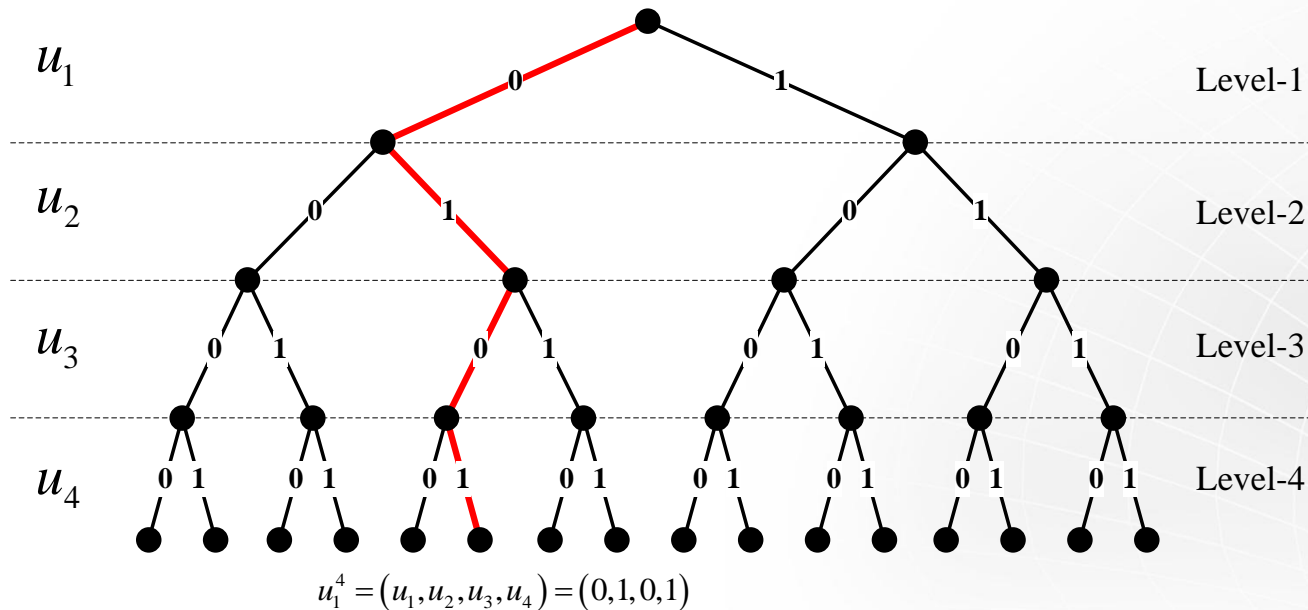
What is the TREE for polar decoding?



- Any polar code with length N is related to a full binary tree with depth N ;
- Each level of the edges is related to an information or a frozen bit;
- Except the leaf nodes, the two dependants of each node are respectively labeled as 0 and 1;
- Each path from the root to any leaf node is related to a decoding sequence (frozen bit included).

Decoding Algorithm for Polar Codes

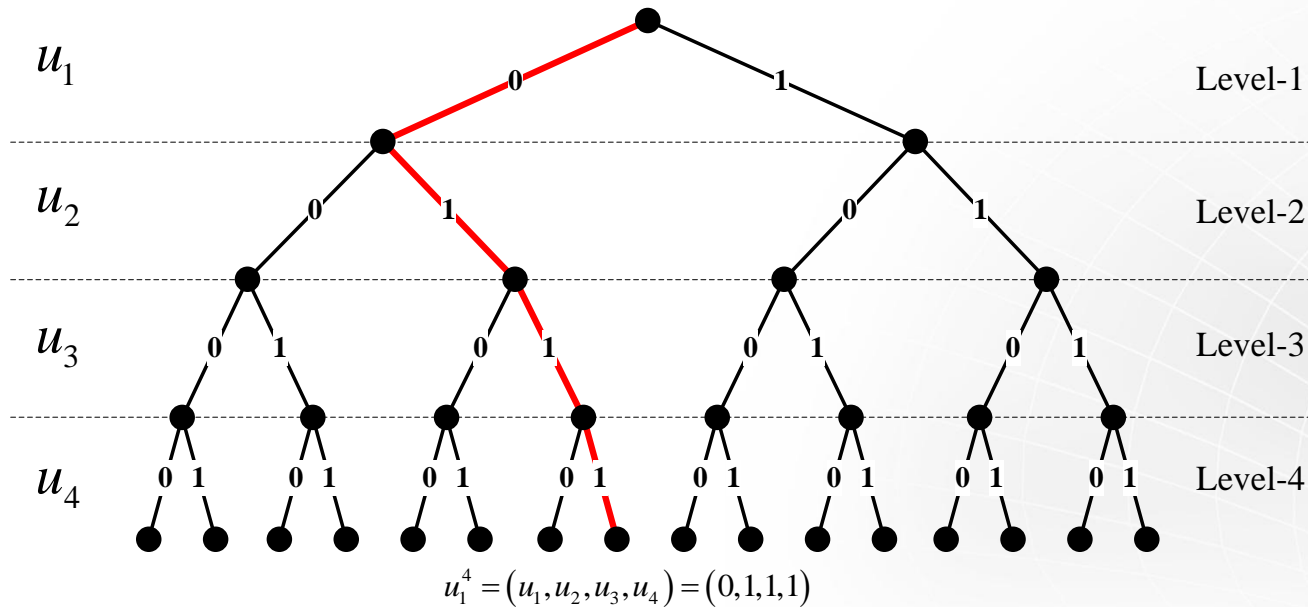
What is the TREE for polar decoding?



- Any polar code with length N is related to a full binary tree with depth N ;
- Each level of the edges is related to an information or a frozen bit;
- Except the leaf nodes, the two dependants of each node are respectively labeled as 0 and 1;
- Each path from the root to any leaf node is related to a decoding sequence (frozen bit included).

Decoding Algorithm for Polar Codes

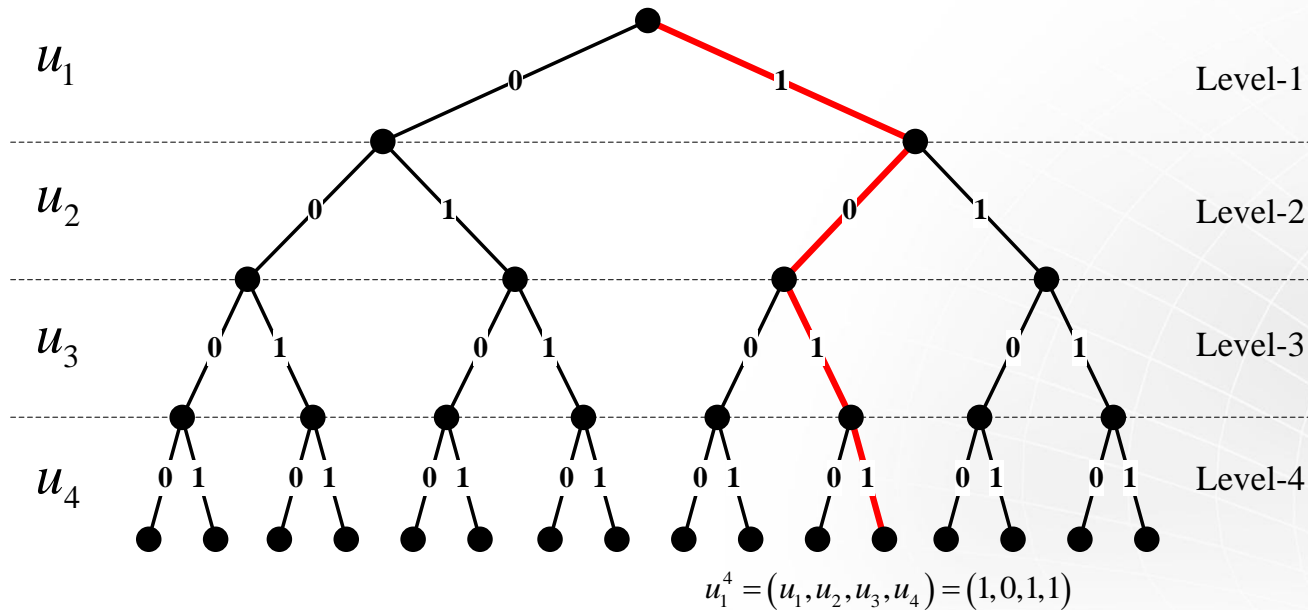
What is the TREE for polar decoding?



- Any polar code with length N is related to a full binary tree with depth N ;
- Each level of the edges is related to an information or a frozen bit;
- Except the leaf nodes, the two dependants of each node are respectively labeled as 0 and 1;
- Each path from the root to any leaf node is related to a decoding sequence (frozen bit included).

Decoding Algorithm for Polar Codes

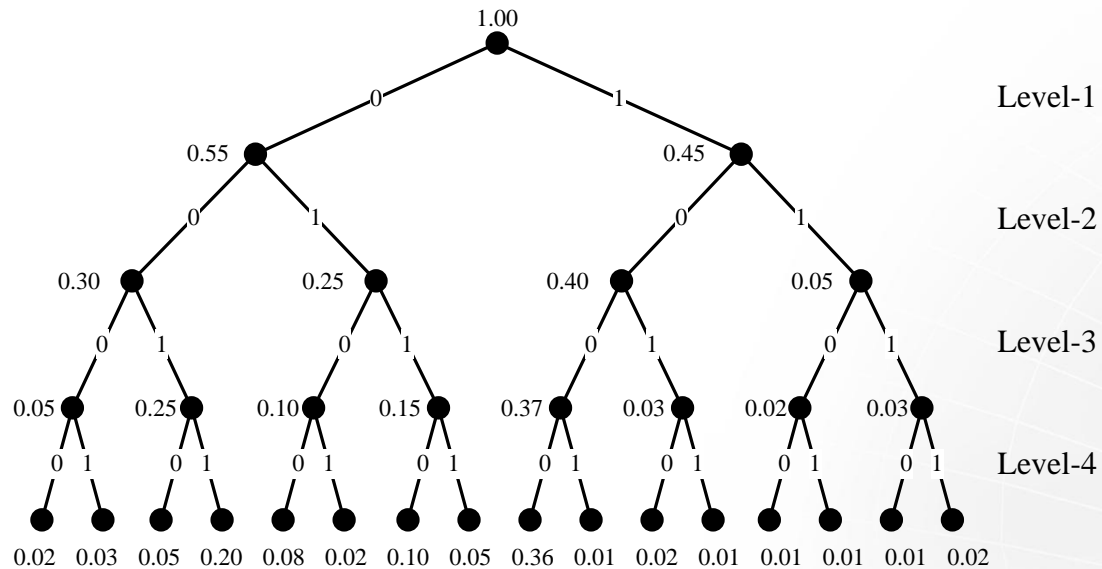
What is the TREE for polar decoding?



- Any polar code with length N is related to a full binary tree with depth N ;
- Each level of the edges is related to an information or a frozen bit;
- Except the leaf nodes, the two dependants of each node are respectively labeled as 0 and 1;
- Each path from the root to any leaf node is related to a decoding sequence (frozen bit included).

Decoding Algorithm for Polar Codes

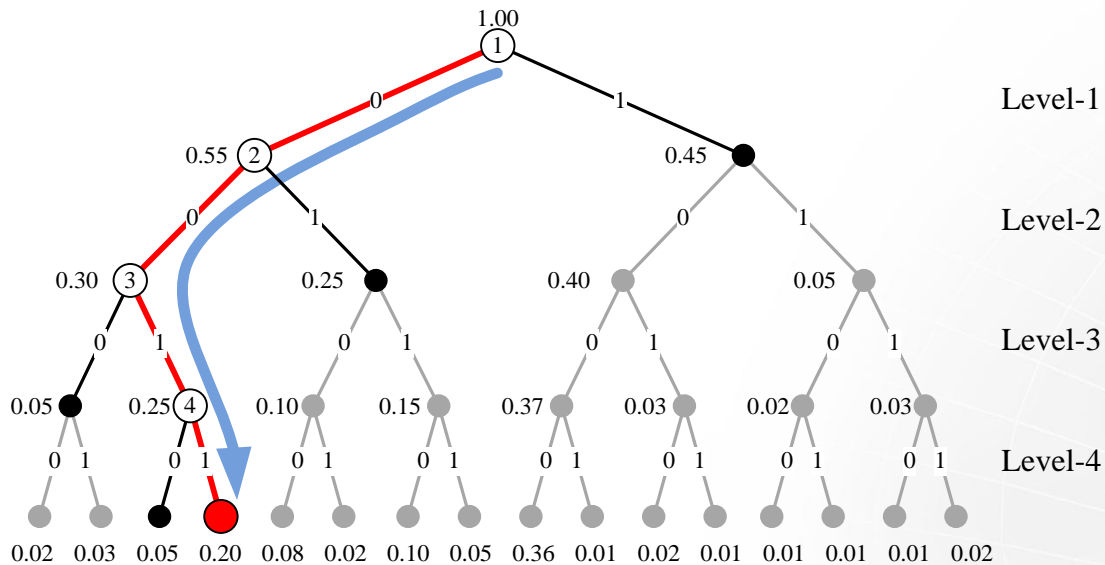
Which paths should be extended?



➤ Each path, from the root to any node, is related to a metric;

Decoding Algorithm for Polar Codes

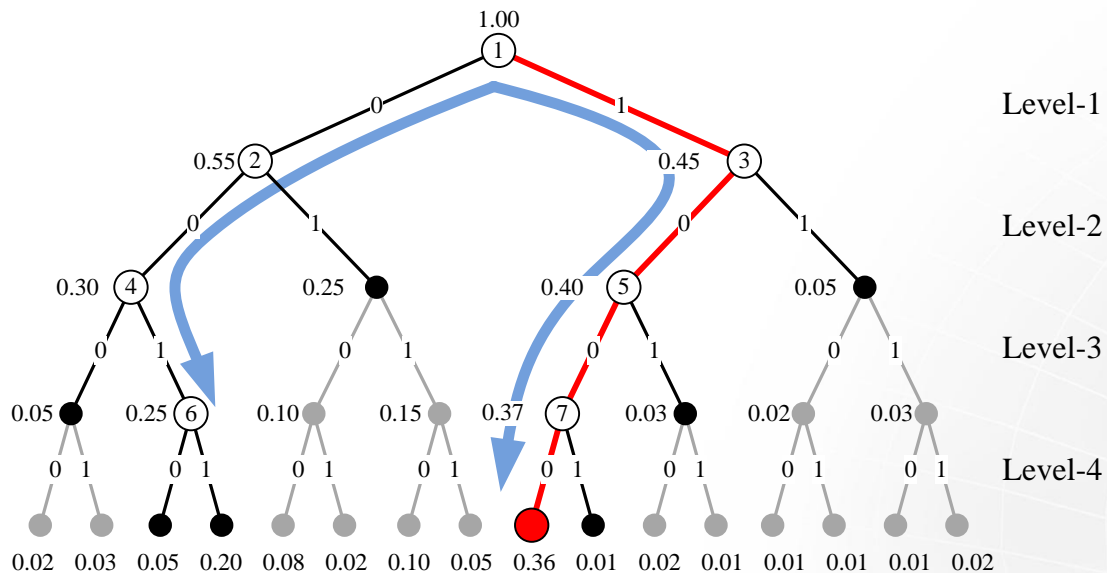
Which paths should be extended?



- Each path, from the root to any node, is related to a metric;
- Starting from the root, do width-first search on the code tree;
- In each level, the **(at most) L paths with larger metrics** are picked out for future expansion, the L is called the search width;

Decoding Algorithm for Polar Codes

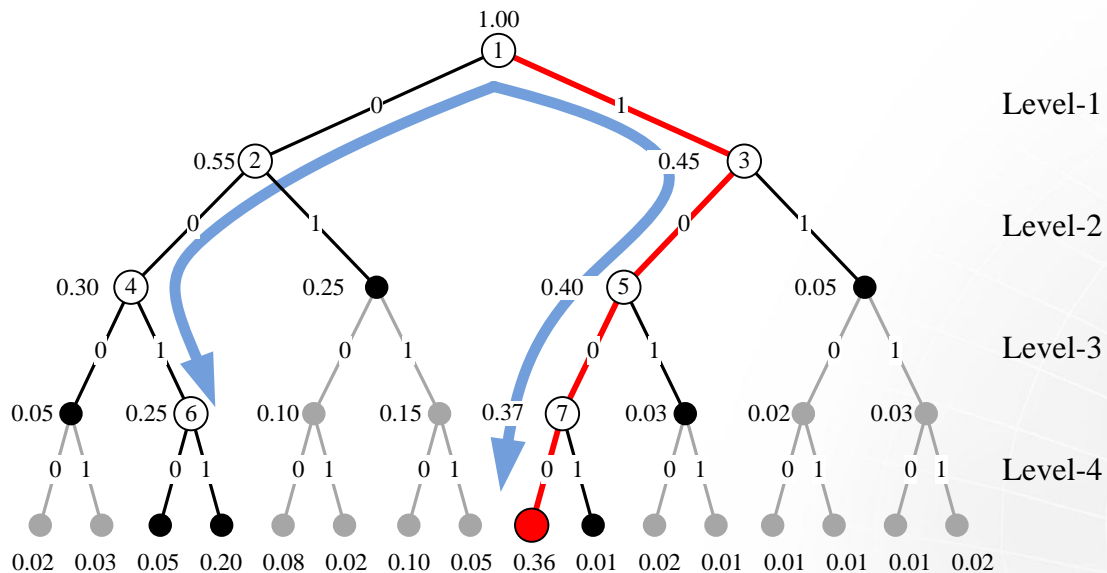
Which paths should be extended?



- Each path, from the root to any node, is related to a metric;
- Starting from the root, do width-first search on the code tree;
- In each level, the **(at most) L paths with larger metrics** are picked out for future expansion, the L is called the search width;

Decoding Algorithm for Polar Codes

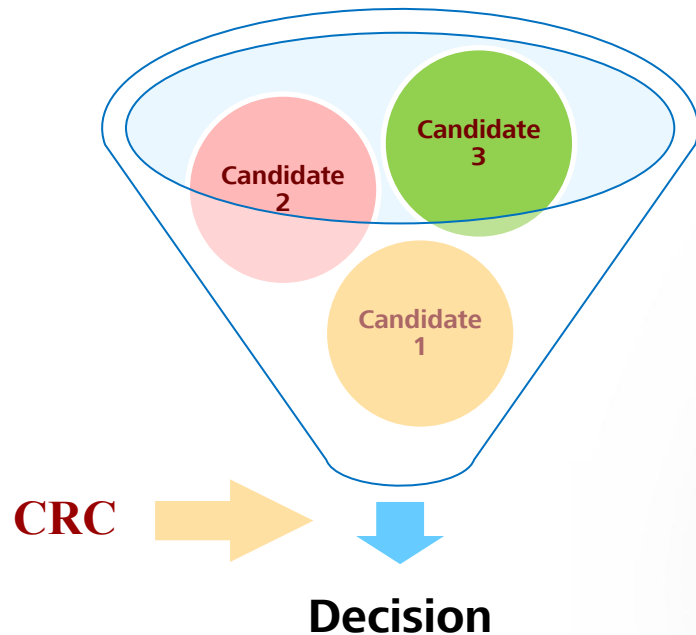
Which paths should be extended?



- Each path, from the root to any node, is related to a metric;
- Starting from the root, do width-first search on the code tree;
- In each level, the **(at most) L paths with larger metrics** are picked out for future expansion, the L is called the search width;
- When achieving the leaf node level, the L candidates are output in the order of decreasing metrics;

Decoding Algorithm for Polar Codes

Which paths should be extended?



- Each path, from the root to any node, is related to a metric;
- Starting from the root, do width-first search on the code tree;
- In each level, the **(at most) L paths with larger metrics** are picked out for future expansion, the L is called the search width;
- When achieving the leaf node level, the L candidates are output in the order of decreasing metrics;
- The one with the largest metric value (can also pass CRC if CA-SCL decoding) are pick out as final decision.

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

- The path metric is defined as the conditional probability of the corresponding bit sequence given the receiving signals, and it is usually in logarithmic form in software/hardware implementation.

$$PM(u_1^i) = \ln(\Pr\{u_1^i | y_1^N\})$$

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

- The path metric is defined as the conditional probability of the corresponding bit sequence given the receiving signals, and it is usually in logarithmic form in software/hardware implementation.

$$PM(u_1^i) = \ln(\Pr\{u_1^i | y_1^N\})$$

- The path metric can be recursively computed as follows:

$$PM(u_1^i) = \begin{cases} PM(u_1^{i-1}) & \text{if } u_i \text{ is information bit or a correct frozen bit, and } (1-2u_i) = \text{sign}(L_N^{(i)}(y_1^N, u_1^{i-1})) \\ PM(u_1^{i-1}) - |L_N^{(i)}(y_1^N, u_1^{i-1})| & \text{if } u_i \text{ is information bit or a correct frozen bit, and } (1-2u_i) \neq \text{sign}(L_N^{(i)}(y_1^N, u_1^{i-1})) \\ -\infty & \text{if } u_i \text{ is a wrong frozen bit} \end{cases}$$

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

- The path metric is defined as the conditional probability of the corresponding bit sequence given the receiving signals, and it is usually in logarithmic form in software/hardware implementation.

$$PM(u_1^i) = \ln(\Pr\{u_1^i | y_1^N\})$$

- The path metric can be recursively computed as follows:

$$PM(u_1^i) = \begin{cases} PM(u_1^{i-1}) & \text{if } u_i \text{ is information bit or a correct frozen bit, and } (1-2u_i) = \text{sign}(L_N^{(i)}(y_1^N, u_1^{i-1})) \\ PM(u_1^{i-1}) - |L_N^{(i)}(y_1^N, u_1^{i-1})| & \text{if } u_i \text{ is information bit or a correct frozen bit, and } (1-2u_i) \neq \text{sign}(L_N^{(i)}(y_1^N, u_1^{i-1})) \\ -\infty & \text{if } u_i \text{ is a wrong frozen bit} \end{cases}$$

where, $PM(\phi) = 0$

$$L_{2N}^{(2i-1)}(y_1^{2N}, \hat{u}_1^{2i-2}) = \text{sign}(L_1 \cdot L_2) \cdot \min(|L_1|, |L_2|)$$

$$L_{2N}^{(2i)}(y_1^{2N}, \hat{u}_1^{2i-1}) = (-1)^{\hat{u}_{2i-1}} \cdot L_1 + L_2$$

$$\begin{cases} L_1 = L_N^{(i)}(y_1^N, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) \\ L_2 = L_N^{(i)}(y_{N+1}^{2N}, \hat{u}_{1,e}^{(2i-2)}) \end{cases}$$



$$L_1^{(1)}(y_i) = \ln \frac{\Pr\{x_i = 0 | y_i\}}{\Pr\{x_i = 1 | y_i\}}$$

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

- The path metric is defined as the conditional probability of the corresponding bit sequence given the receiving signals, and it is usually in logarithmic form in software/hardware implementation.

$$PM(u_1^i) = \ln(\Pr\{u_1^i | y_1^N\})$$

- The path metric can be recursively computed as follows:

$$PM(u_1^i) = \begin{cases} PM(u_1^{i-1}) & \text{if } u_i \text{ is information bit or a correct frozen bit, and } (1-2u_i) = \text{sign}(L_N^{(i)}(y_1^N, u_1^{i-1})) \\ PM(u_1^{i-1}) - |L_N^{(i)}(y_1^N, u_1^{i-1})| & \text{if } u_i \text{ is information bit or a correct frozen bit, and } (1-2u_i) \neq \text{sign}(L_N^{(i)}(y_1^N, u_1^{i-1})) \\ -\infty & \text{if } u_i \text{ is a wrong frozen bit} \end{cases}$$

where, $PM(\phi) = 0$

$$\begin{cases} L_{2N}^{(2i-1)}(y_1^{2N}, \hat{u}_1^{2i-2}) = \text{sign}(L_1 \cdot L_2) \cdot \min(|L_1|, |L_2|) \\ L_{2N}^{(2i)}(y_1^{2N}, \hat{u}_1^{2i-1}) = (-1)^{\hat{u}_{2i-1}} \cdot L_1 + L_2 \end{cases} \quad \begin{cases} L_1 = L_N^{(i)}(y_1^N, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) \\ L_2 = L_N^{(i)}(y_{N+1}^{2N}, \hat{u}_{1,e}^{(2i-2)}) \end{cases} \quad L_1^{(1)}(y_i) = \ln \frac{\Pr\{x_i = 0 | y_i\}}{\Pr\{x_i = 1 | y_i\}}$$

Given u_1^N , the sub-vectors consists of elements with odd and even indices are denoted by $u_{1,o}^N$ and $u_{1,e}^N$

$$u_{1,o}^N = (u_1, u_3, \dots, u_{2k-1}, \dots, u_{N-1}) \quad u_{1,e}^N = (u_2, u_4, \dots, u_{2k}, \dots, u_N)$$

Decoding Algorithm for Polar Codes

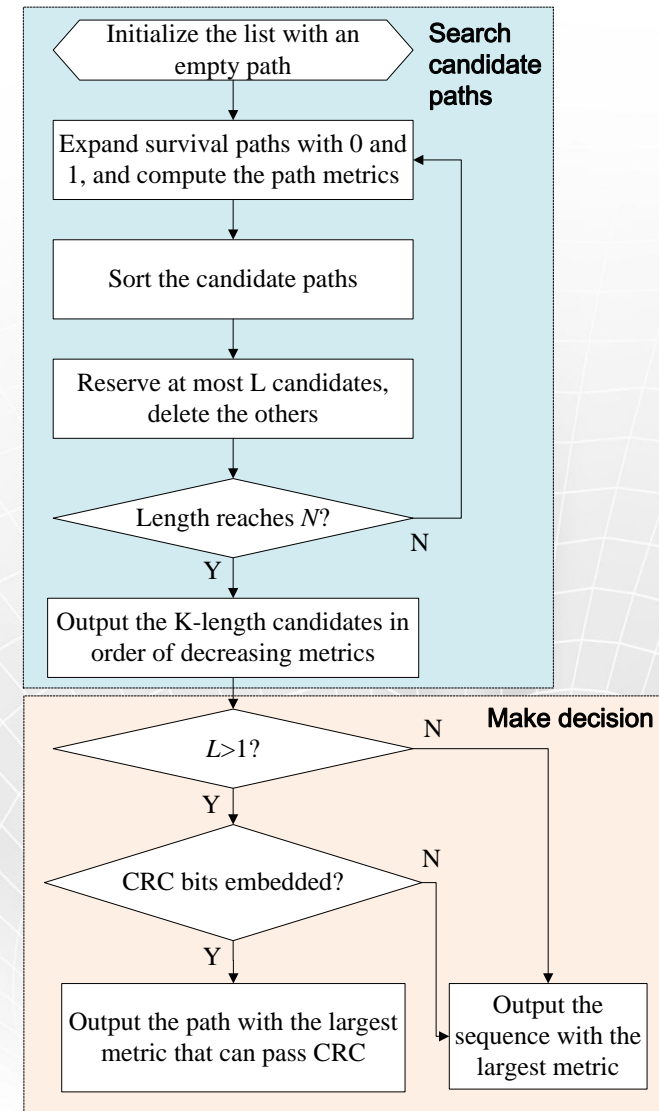
Summary

➤ Decoder input

N	Code length, the length of encoded sequence, take value as some power of 2
y_1^N	Receiving signal from BPSK modulated AWGN channel
A	Set of information bit indices, size is K
L	Search width of SCL decoding algorithm
σ^2	Variance of AWGNs

➤ Decoder output

u_A	K -length information bit sequence
-------	--------------------------------------



How polar code will be implemented for this task

IMPLEMENTATION GUIDE

Phase I

■ REQUIREMENTS

- Correctly understand the polar encoding and decoding algorithm, simulate the BER/BLER performance;
- Design FPGA structures of polar encoder and decoder, evaluate the resources, data rate and delay;

■ DELIVERIES

- Algorithm document: BER/BLER performance under BPSK modulated AWGN channel;
- Simulation code writing in C or Matlab;
- FPGA outline design document;

■ CRITERIA

- Correctly understand the polar encoding and decoding algorithms; the software simulation results is reasonable;
- Good FPGA design for polar encoder and decoder.

Phase II

■ REQUIREMENTS

- Implement the polar encoding & decoding algorithms using hardware description language (Verilog RTL etc.);
- Implement hardware polar encoder and decoder based on the provided platform DE5-Net Dev Kit;

■ DELIVERIES

- FPGA design document;
- The source code and FPGA bit stream (.bit) file;
- Test result of the FPGA implementation;

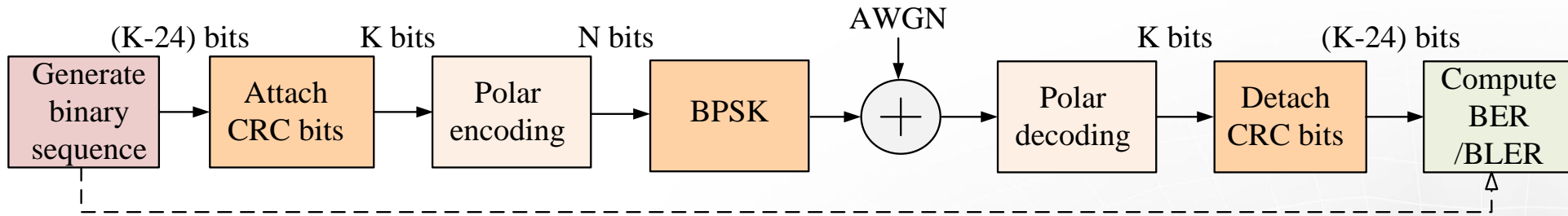
■ CRITERIA

- Complete the FPGA implementation of polar encoder and decoder;
- High data throughput, low processing delay and less FPGA resources occupation is desired;
- Novel algorithm and architecture is a plus.

System Configuration Parameters for Implementation

Parameter	Value	Description
N	256 or 1024	The code length is set to 256 or 1024; The simulation should support both code length configurations; In FPGA design, competitors can chose one to implement.
K	128 or 512	Length of information blocks: when $N=256$, $K=128$; when $N=1024$, $K=512$; 24 CRC bits are assumed to be embedded in the info. block;
$g(D)$	From LTE	The generator polynomial is $g(D)=D^{24}+D^{23}+D^{18}+D^{17}+D^{14}+D^{11}+D^{10}+D^7+D^6+D^5+D^4+D^3+D+1$
L	1 to 32	Searching width, usually be some power of 2, e.g. 1, 2, 4, 8, 16, 32; For hardware design, competitors can chose one to implement.
A	Indicated by 1s in next two pages	Set of information bit indices, known at both encoder and decoder; sized K
u_{A^c}	All zeros	Frozen bit sequences, with length $(N-K)=128/512$, known at both encoder and decoder

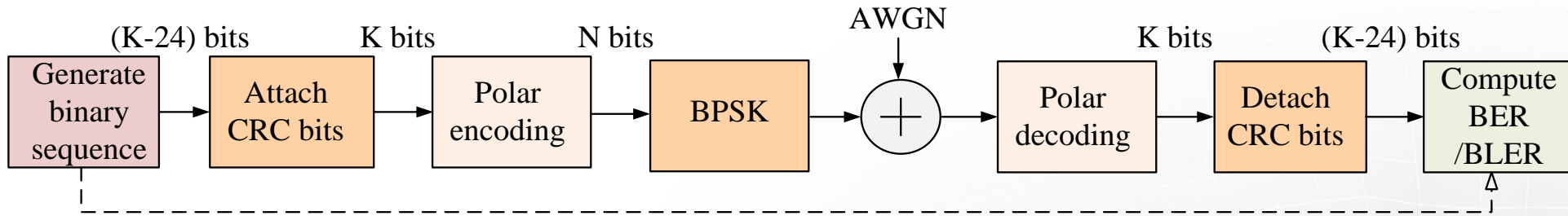
System Diagram for Simulation and FPGA



Block Diagram for Polar Code Simulation

- The source message is (K-24) length binary sequences, with bit 0 and bit 1 uniformly distributed;
- The K-length input sequence of polar encoder includes 24 CRC bits;
- Demodulation is merged into the polar decoder;
- The decoding results is compared with the source sequence, and compute BER and BLER:
 - BER (Bit Error Rate) is the number of error bits divided by the total number of bits transmitted bit;
 - BLER (Block Error Rate) is the number of error blocks divided by the total number of sequences sent.

System Diagram for Simulation and FPGA



Block Diagram for Polar Code Simulation

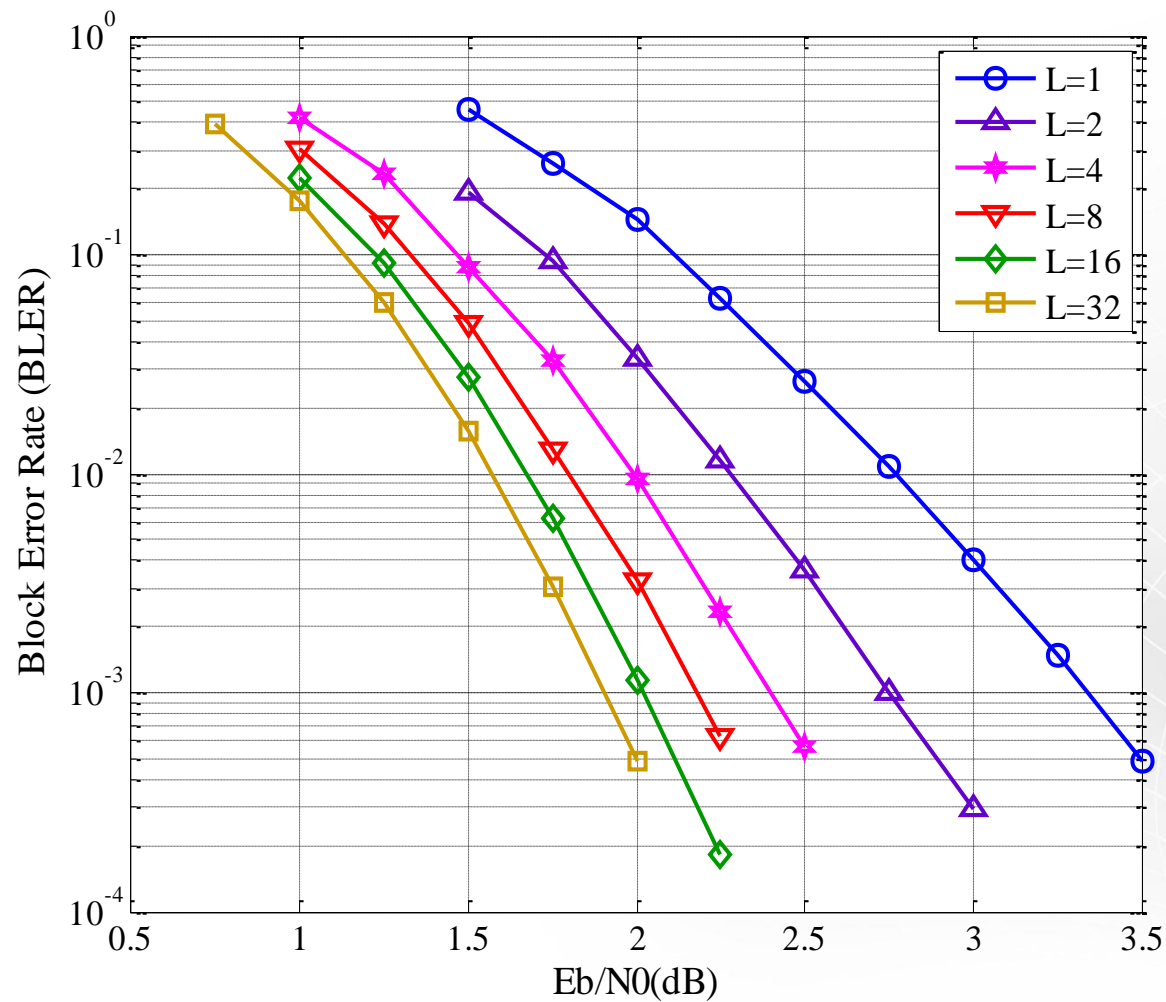
- BPSK maps bit 0/1 to real number signals +1/-1: $\begin{cases} 0 \rightarrow +1 \\ 1 \rightarrow -1 \end{cases}$
- AWGN is a Gaussian distributed variable with average 0, and variance σ^2 ;
 - E.g., AWGN sequence with length N and variance $\sigma^2 = 0.64$ can be generated using Matlab code as

`noises = sqrt(0.64)*randn(1, N);`

- Noise variance is calculated based on the target bit SNR (E_b / N_0) for simulation:

$$E_b / N_0 \text{ (dB)} = \log_{10} \frac{NE_b}{2(K - L_{CRC})\sigma^2} = 10 \times \log_{10} \frac{1024}{2 \times (512 - 24)\sigma^2} \approx 10 \times \log_{10} \frac{1.0492}{\sigma^2} \text{ dB}$$

Performance Curves for Reference



- BPSK+AWGN
- $N=1024, K=512$
- CA-SCL decoding
- Float-point implementation
- #Simulated frames $> 10^5$

Hints on the Polar Decoder Implementation

■ Memory storage for candidate path

- For a direct implementation of polar decoder, storing each path cost a space complexity $O(N \log N)$;
- Utilizing features of SC/SCL, only a space complexity $O(N)$ is required for each path.

■ Path arrangement in memory

- A large amount of memory copy operation are needed in SCL decoding;
- A direct implementation of these copies will cost $O(N^2)$ time complexity;
- By introducing data sharing between paths, only the parts need to be updated will be copied, the complexity can be significantly reduced (it is called ‘lazy copy’);

■ Quantization scheme

- The quantization scheme of LLRs and path metrics has significant impact on the performance;
- A recommended solution is to use more than 8 bit for LLRs and 10 bit for metrics with 2 bit after decimal point;
- The path metric take values as non-positive values, so the sign bit is not a must.

[1] I. Tal and A. Vardy, "List decoding of polar codes," arXiv:1206.0050v1, 2012 [Online]. Available: <http://arxiv.org/pdf/1206.0050v1>

ALTERA.
UNIVERSITY
PROGRAM

THANK YOU

Organizer |



西安电子科技大学
XIDIAN UNIVERSITY



Sponsor |

