

ALTERA.
UNIVERSITY
PROGRAM

5G

5G第一届移动通信算法创新大赛
赛题及要求

Polar Code

主办单位 |



西安电子科技大学
XIDIAN UNIVERSITY

terasic Terasic Inc.

赞助厂商 |



SPREADTRUM

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

Task	Description	Requirements
Polar code – 能够达到信道容量极限的高性能纠错码	<p>自现代通信技术诞生以来，以更低的代价（信号功率、信号带宽等）实现更可靠的通信一直是信息技术领域的核心课题。信道编码技术是无线通信系统物理层最核心的基础技术之一，它的主要目的是使数字信号能够进行可靠的传递。信道编码技术通过在发送信息序列的基础上增加额外的校验比特，并在收端采用一定的译码技术以较高的概率对传输过程中产生的差错进行纠正，从而实现发送信息序列的正确接收。</p> <p>为了实现可靠的信号传输，编码学家在过去的半个多世纪提出多种纠错码技术如RS码、卷积码、Turbo码、LDPC码等，并在各种通信系统中取得了广泛的应用。但是以往所有实用的编码方法都未能到香农于1948年所给出的信道传输的容量极限（也称为香农界）。2008年在国际信息论ISIT会议上，Arikan首次提出了信道极化的概念，基于该理论，他给出了人类已知的第一种能够被严格证明达到信道容量的信道编码方法，并命名为极化码（polar code）。Polar码具有明确而简单的编码及译码算法。通过信道编码学者的不断努力，当前Polar码所能达到的纠错性能超过目前广泛使用的Turbo码、LDPC码。</p> <p>请根据本文所提供的材料以及公开发表的参考文献，理解Polar码编译码算法，并设计编译码程序，进行性能仿真，输出正确的仿真结果。基于Polar码编译码算法，提出一种Polar码FPGA逻辑实现架构，并采用Verilog RTL等硬件描述语言进行编程实现，利用本次大赛所提供的硬件及工具，进行FPGA测试，并输出相应的测试结果。</p>	<p>要求：</p> <p>根据本赛所提供Polar码的介绍材料及参考文献，实现Polar码编译码算法，重点在Polar码编码和译码模块的开发、验证及测试。</p> <p>作品格式：</p> <ol style="list-style-type: none">理解Polar编译码算法啊，设计Polar码编译码程序，进行性能仿真，并输出算法仿真结果；基于Polar码编译码算法，设计一种Polar码FPGA逻辑实现架构，并采用Verilog RTL等硬件描述语言进行编程实现，利用本次大赛所提供的硬件及工具，进行FPGA测试，并输出相应的测试结果。 <p>交付材料：</p> <ol style="list-style-type: none">Polar码编译码算法程序设计说明书、算法程序（C或Matlab），及误比特率（BER）、误码块率（BLER）仿真结果；Polar码编译器FPGA设计说明书，硬件描述程序、比特文件及测试结果。 <p>评选标准：</p> <p>初赛：</p> <ol style="list-style-type: none">正确理解Polar码编译码算法，正确编写C/Matlab算法程序及输出正确的仿真结果提交的Polar码编译码算法链路FPGA设计架构 <p>复赛：</p> <ol style="list-style-type: none">成功完成Polar码编译码算法链路的FPGA开发及测试FPGA设计所实现的算法、数据吞吐率、处理时延及使用的FPGA芯片资源

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>
- 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?



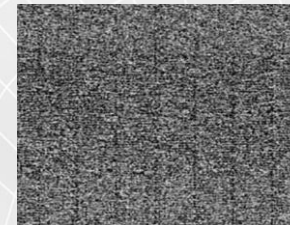
100101.....0
K比特发送信号



发射端

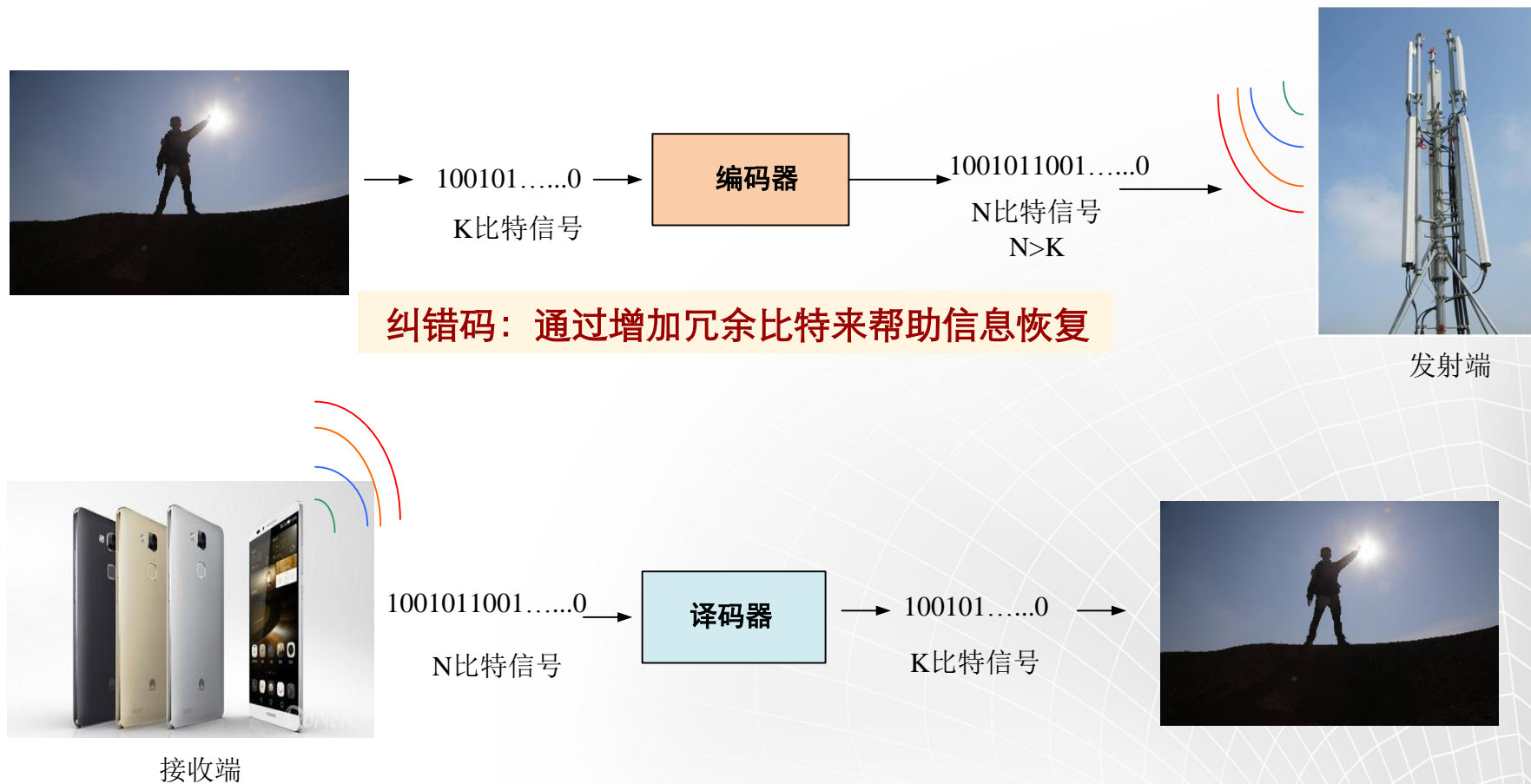


10?10?.....?
K比特信号

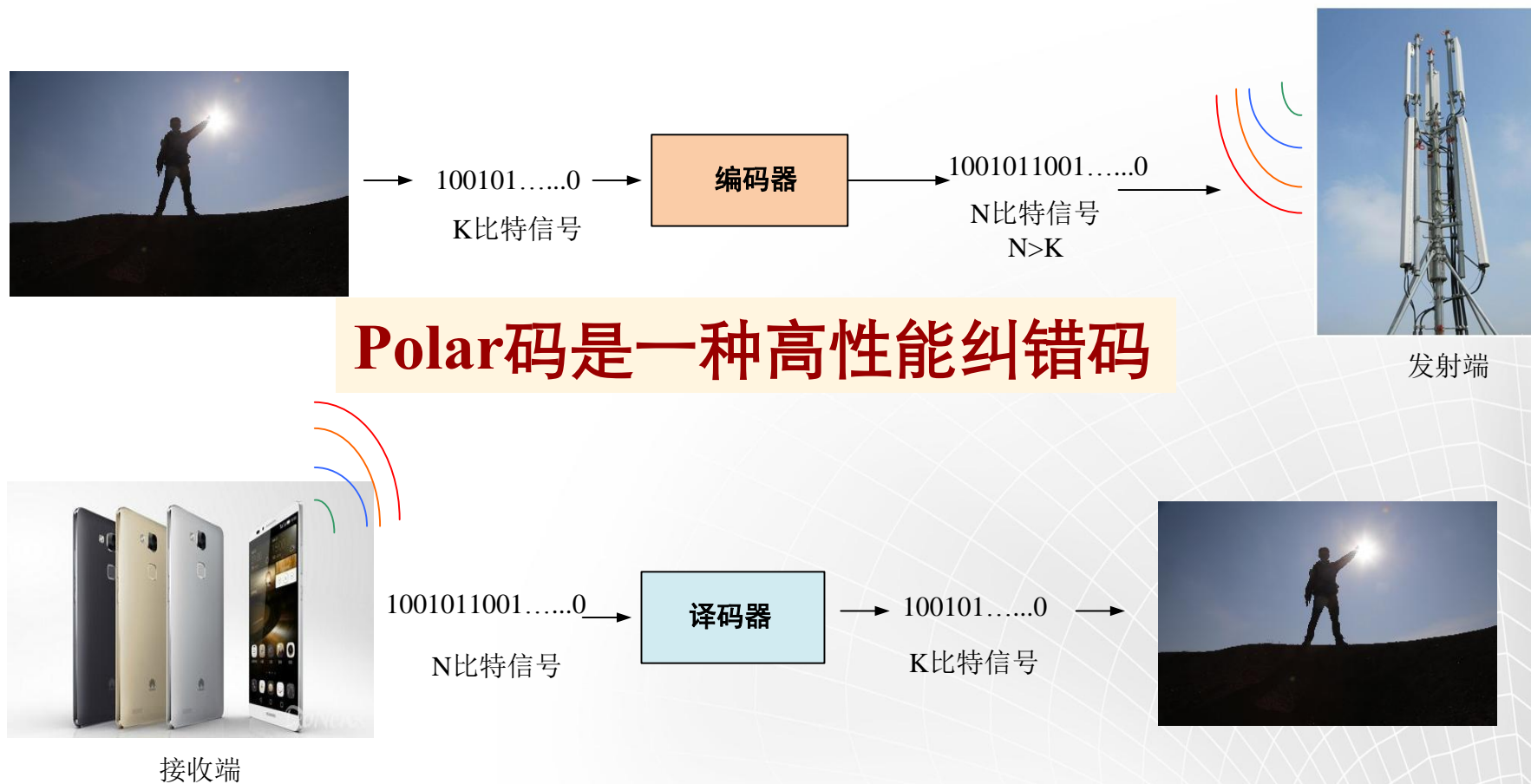


接收端

What is Polar Code?



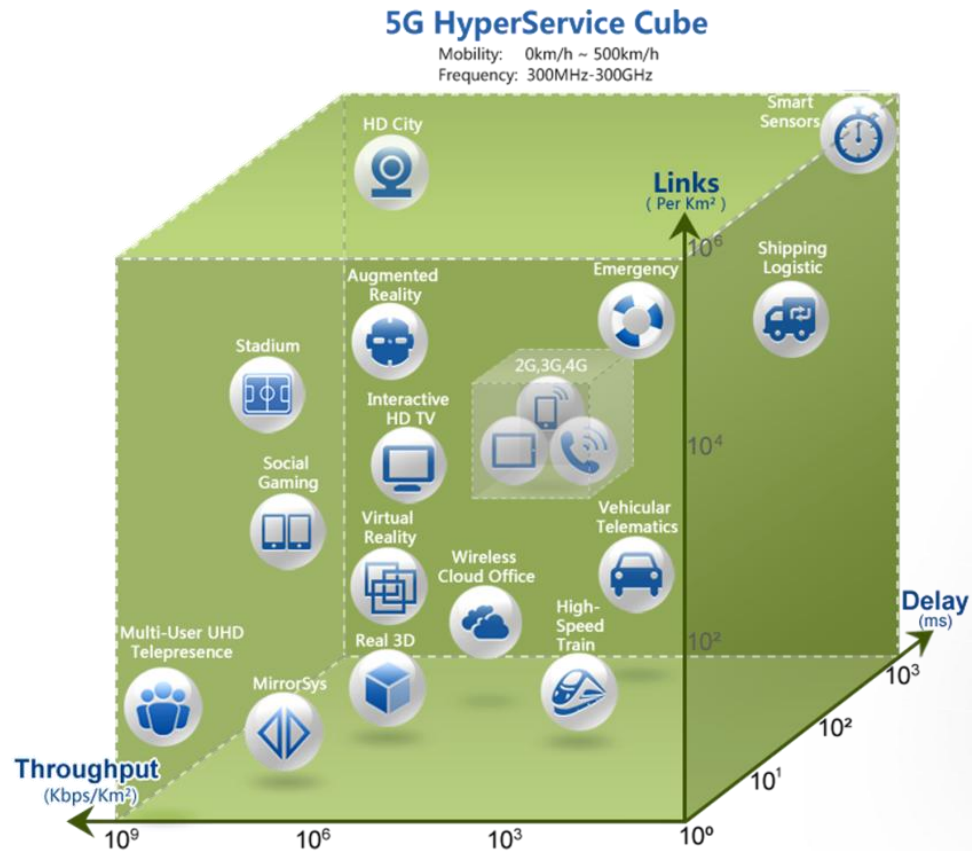
What is Polar Code?



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**

➤ 终端用户数据速率: **10Gbps**

- iPhone, iPad, iGlass, iWatch

➤ 基站数据速率: **1Tbps**

- Cloud computing blade

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

➤ 传感器应用: **10~100Bytes**

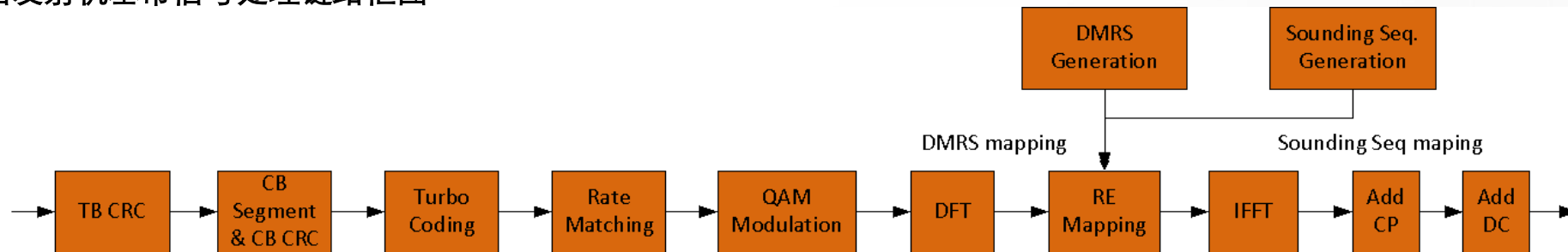
- meters, telemetric, RFID,

➤ 工业控制应用: **10^{-4} second latency**

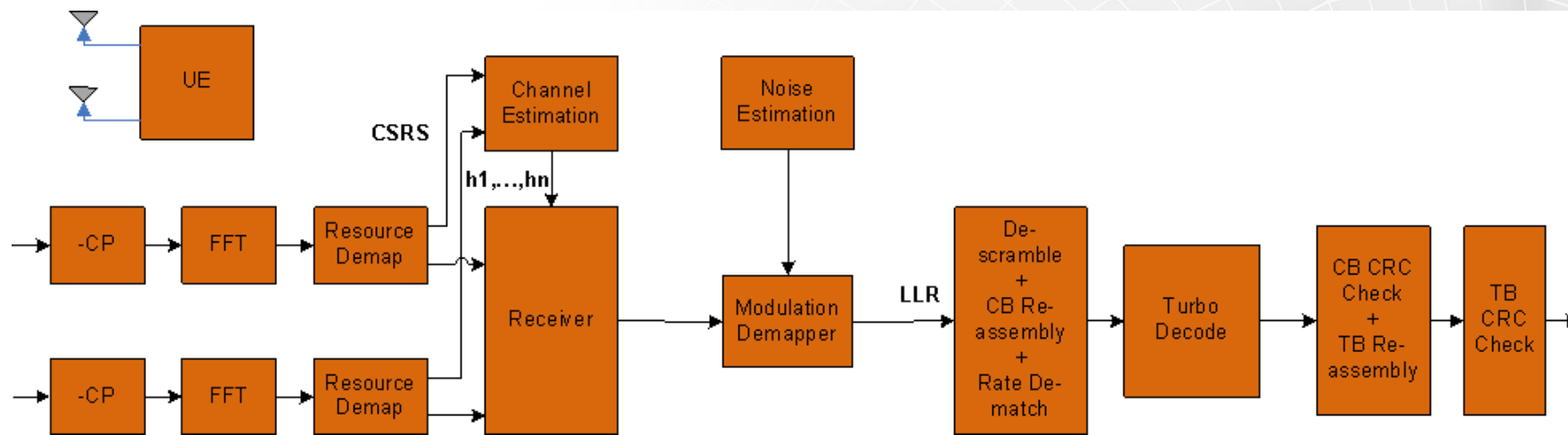
- Could-drive-car, factory control

Polar Code in Wireless System

无线通信发射机基带信号处理链路框图

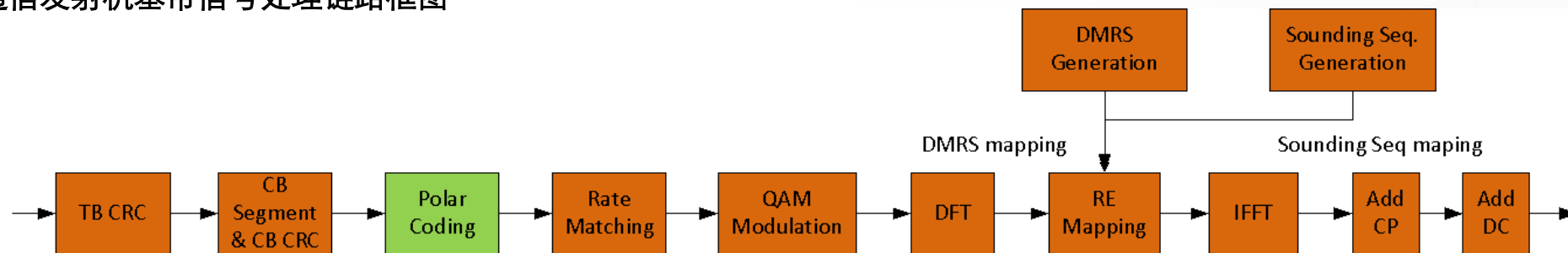


无线通信接收机基带信号处理链路框图

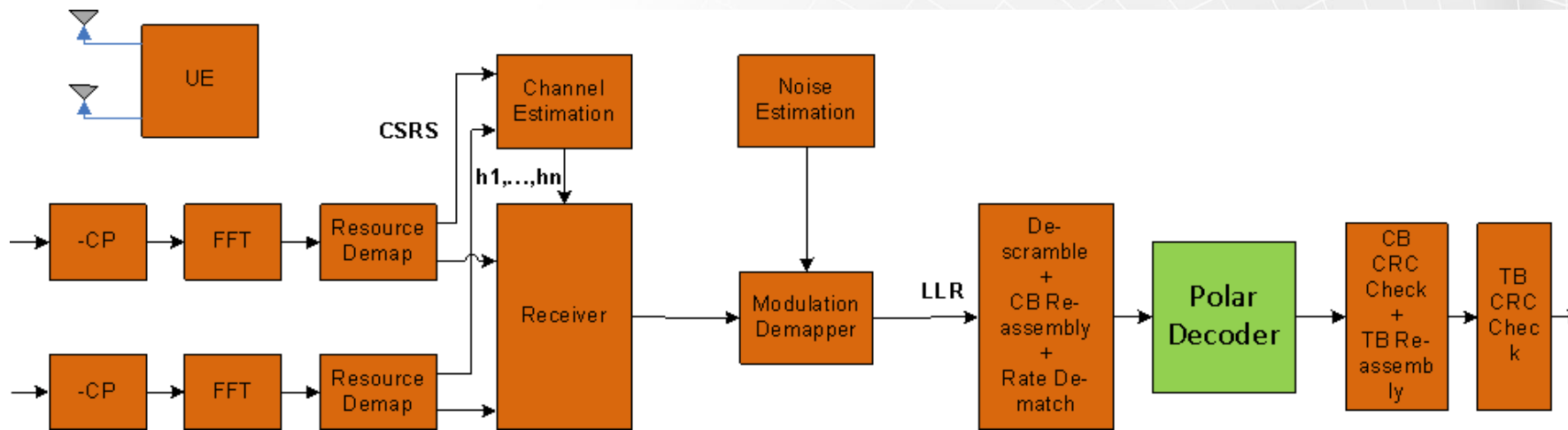


Polar Code in Wireless System

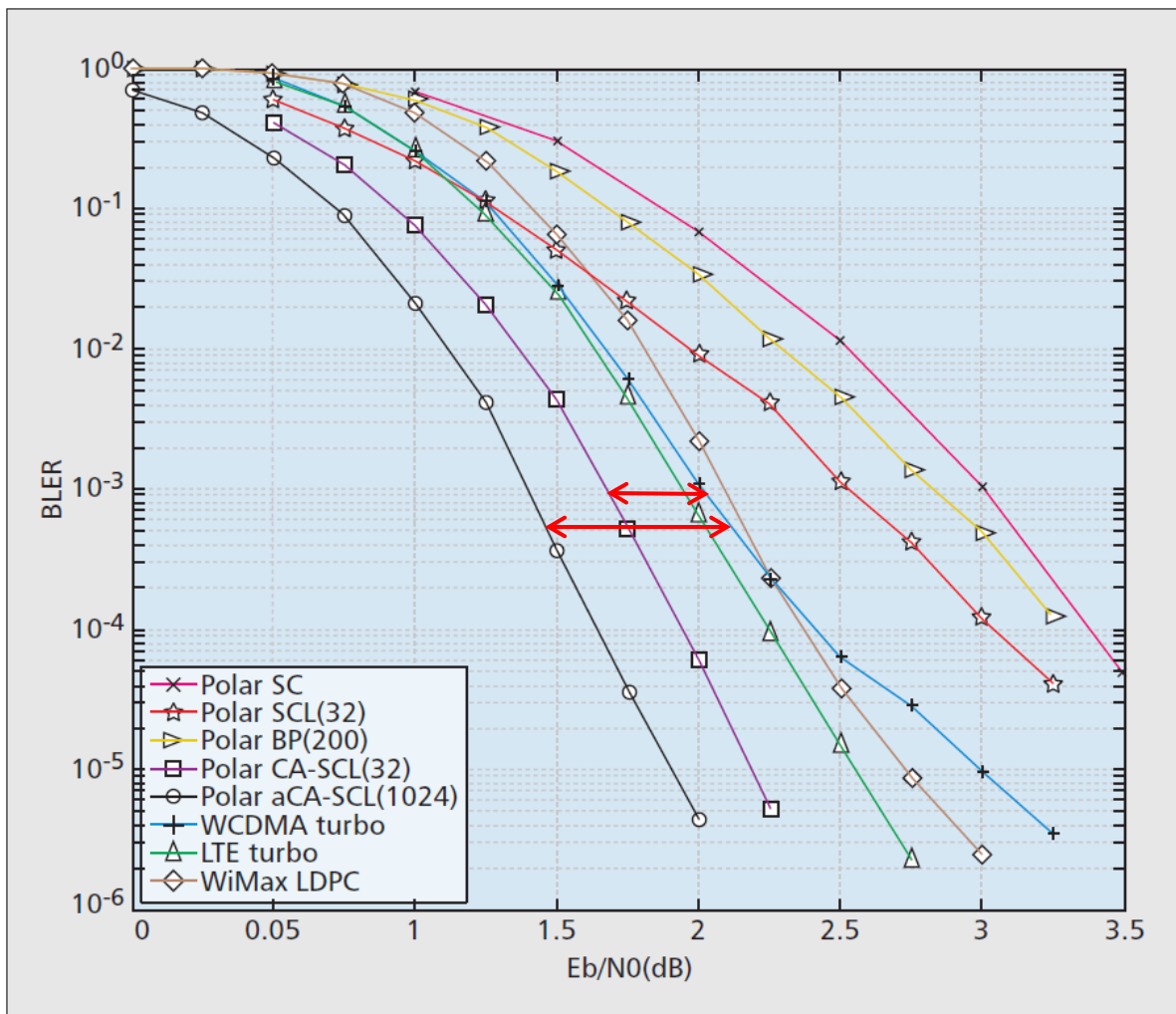
无线通信发射机基带信号处理链路框图



无线通信接收机基带信号处理链路框图



Polar Code vs Classical Code



- 配置：码长1024，码率0.5
- 信道：BPSK+AWGN
- 相比Turbo/LDPC码有0.3~0.7dB的增益

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

HOW DOES POLAR CODE WORK?

Polar Encoding

- 混合信息比特与固定比特得到 $u_1^N = (u_1, u_2, \dots, u_N)$
 - 待编码序列为长度为 K 的信息比特序列;
 - 给定一个对编、译码器均已知的固定比特序列, 长度为 $N-K$;
 - 混合两个序列, 得到 u_1^N : 其中 u_A 为信息比特序列, u_{A^c} 为固定比特序列, 信息比特和固定比特的序号用 A 及其补集 A^c 表示。

- 乘以生成矩阵

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

其中, $n = \log_2 N$, $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, $F^{\otimes n} = F \otimes F^{\otimes(n-1)}$, $\otimes n$ 表示 n 次

克罗内克 (Kronecker) 幂, 即 n 个矩阵 F 连续做克罗内克积。

- 比特反序重排

对 v_1^N 进行比特反序重排, 得到序列 x_1^N , 即是令 $x_i = v_{\pi(i)}$

比特反序函数 $\pi(i)$ 定义如下: 令 i 用二进制表示为 (b_1, b_2, \dots, b_n) ,

$$i = \sum_{k=1}^n (b_k \cdot 2^{n-k}) + 1$$

则 $\pi(i)$ 的值对应的二进制表示为 $(b_n, b_{n-1}, \dots, b_1)$ 。

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

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

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

Polar Encoding

示例：Polar码编码 $N=8$ $K=4$

信息比特序号集合 $A = \{4, 6, 7, 8\}$

固定比特序号集合 $A^c = \{1, 2, 3, 5\}$

信息比特集合为 (i_1, i_2, i_3, i_4)

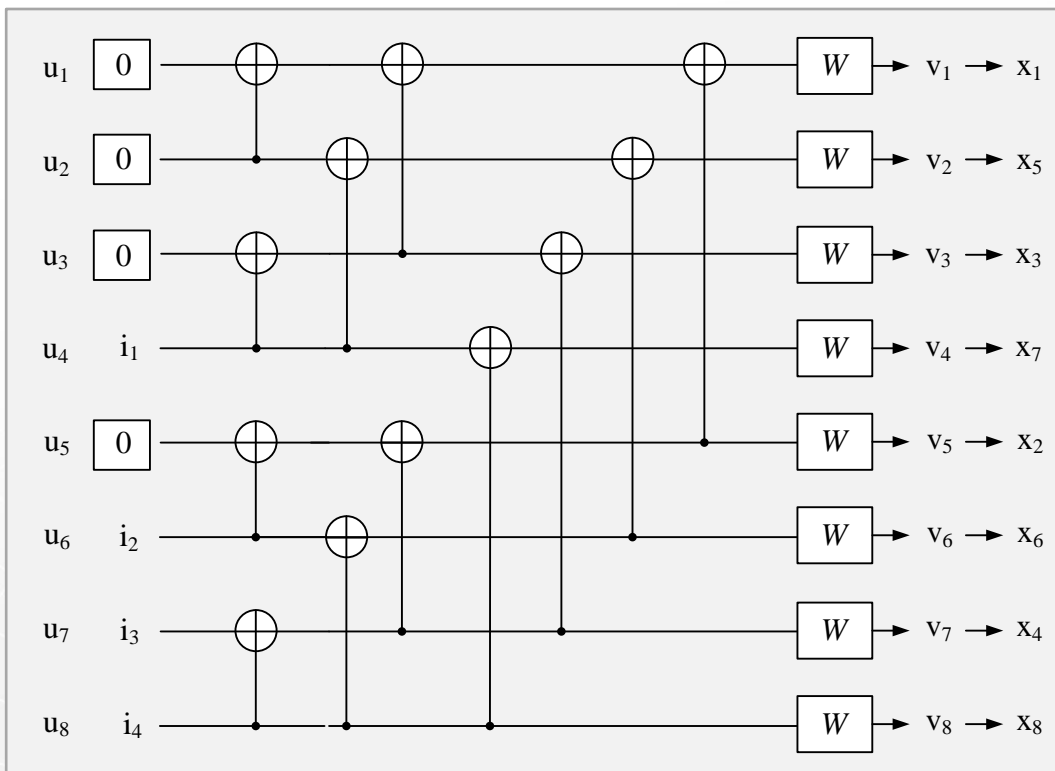
固定比特集合为 $(0, 0, 0, 0)$

混合后，得到 $u_1^8 = (0, 0, 0, i_1, 0, i_2, i_3, i_4)$

$v_1^8 = u_1^8 F^{\otimes 3}$ 可以通过右图结构实现

$(1, 2, 3, 4, 5, 6, 7, 8)$ 的比特反序为 $(1, 5, 3, 7, 2, 6, 4, 8)$

根据 v_1^8 比特反序重排后得到编码比特序列 x_1^8



Decoding Algorithm for Polar Codes

■ Polar码在被提出之初——串行抵消（Success Cancellation, SC）译码

- 复杂度低、译码结构简单
- 理论上被证明在码长足够大时能够达到Shannon极限
- SC译码算法在码长为有限长的配置下，纠错性能不理想

■ 置信度传播（BP）等译码算法性能亦不理想

■ 串行抵消列表（Successive Cancellation List, SCL）译码

- 改进的SC译码算法
- 以较低复杂度代价，获得最大似然译码性能

■ CRC辅助的SCL（CRC-Aided SCL, CA-SCL）译码算法

- 通常信息比特序列均包含有CRC比特
- 利用“正确序列能够通过CRC校验”这一先验信息，对SCL译码算法得到的候选序列集合进行选择；
- 获得较已有其它编码方式（Turbo、LDPC）相当、甚至更优的性能

Decoding Algorithm for Polar Codes



树搜索算法



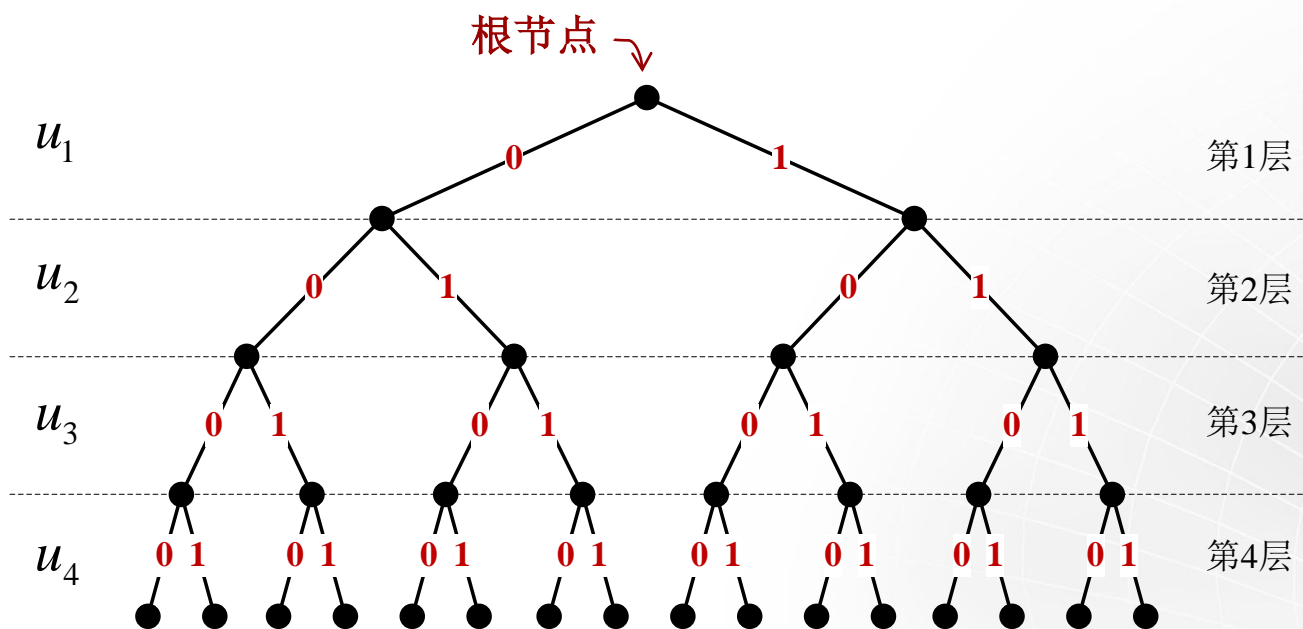
What is the TREE for polar decoding? 译码树的定义

Which paths should be extended? 路径扩展规则

How to calculate the path metrics? 度量值计算

Decoding Algorithm for Polar Codes

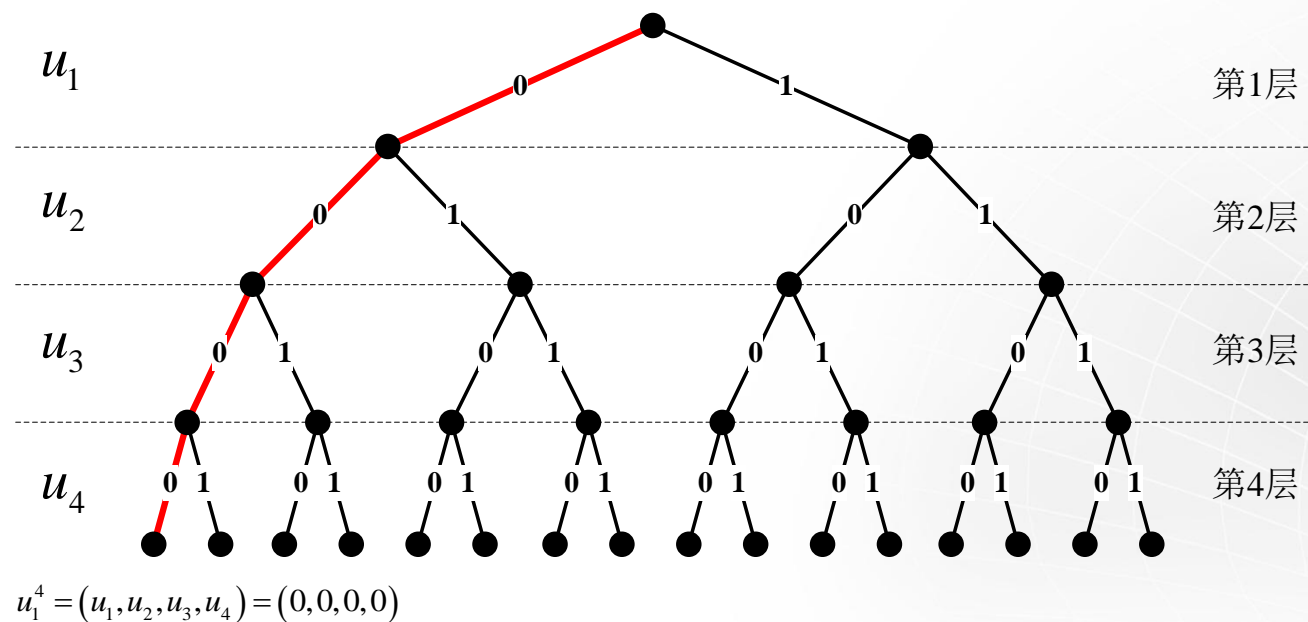
What is the TREE for polar decoding?



- 码长为N 的Polar码，都对应一棵深度为N 的满二叉树；
- 每一层边都分别对应一个信息比特或固定比特；
- 除叶节点外，每一个节点与其左、右两个后继节点之间的边分别被标记为0和1；
- 从根节点出发到任一叶节点长度为N的路径均对应一个译码序列（含固定比特）。

Decoding Algorithm for Polar Codes

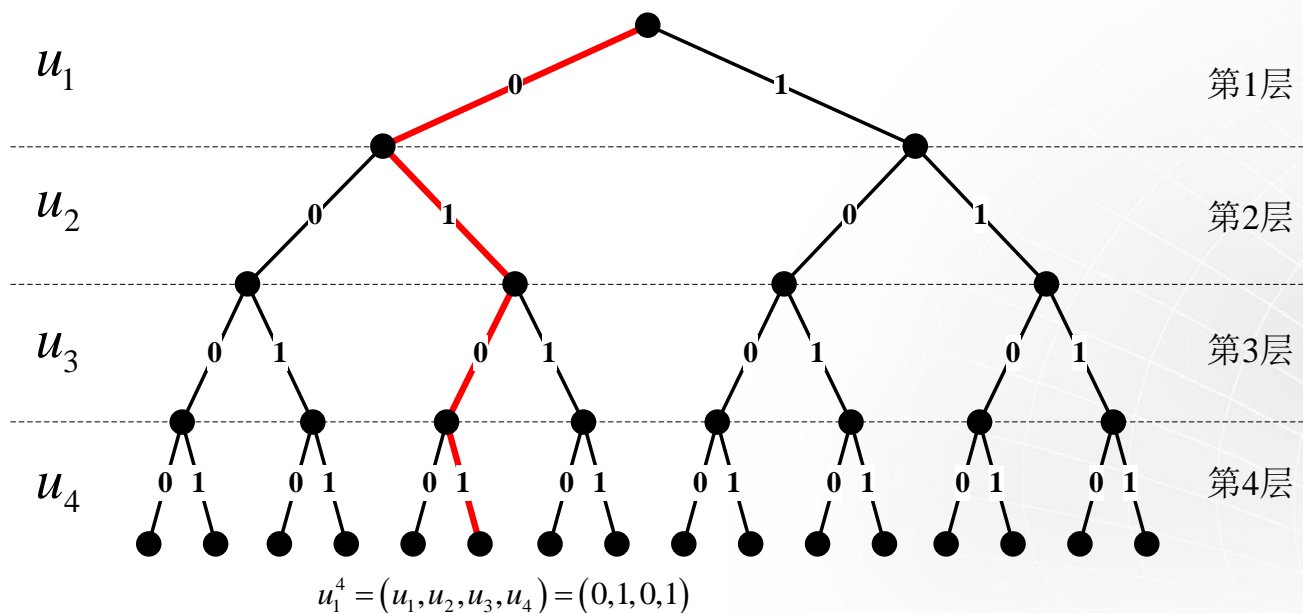
What is the TREE for polar decoding?



- 码长为N 的Polar码，都对应一棵深度为N 的满二叉树；
- 每一层边都分别对应一个信息比特或固定比特；
- 除叶节点外，每一个节点与其左、右两个后继节点之间的边分别被标记为0和1；
- 从根节点出发到任一叶节点长度为N的路径均对应一个译码序列（含固定比特）。

Decoding Algorithm for Polar Codes

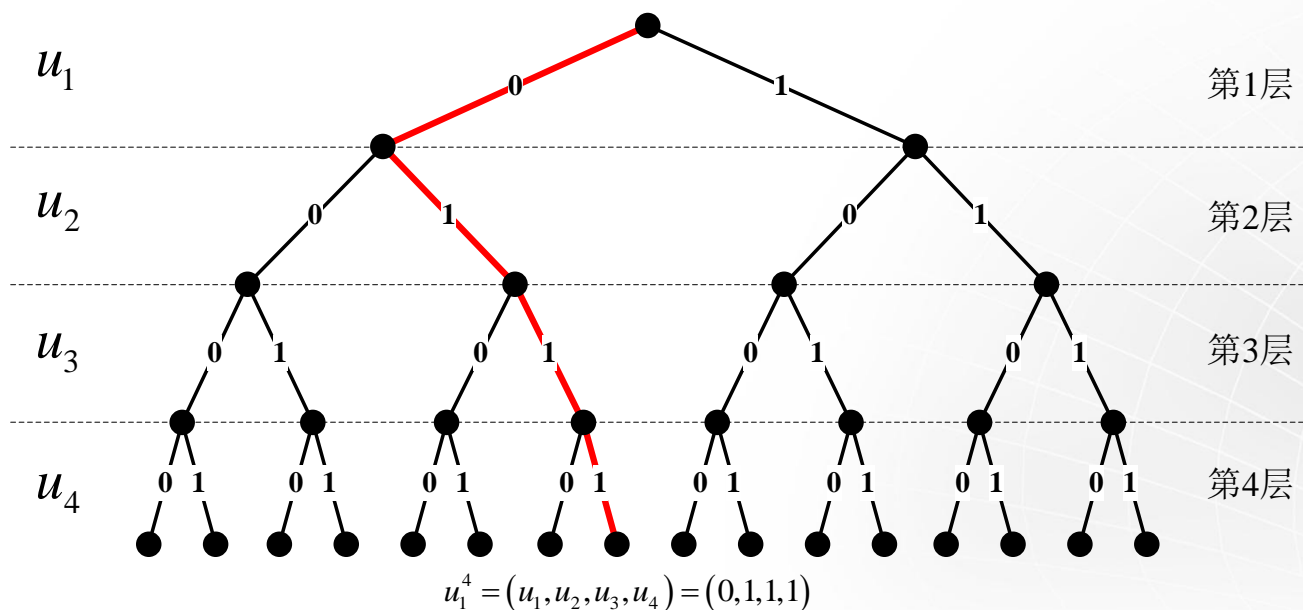
What is the TREE for polar decoding?



- 码长为N 的Polar码，都对应一棵深度为N 的满二叉树；
- 每一层边都分别对应一个信息比特或固定比特；
- 除叶节点外，每一个节点与其左、右两个后继节点之间的边分别被标记为0和1；
- 从根节点出发到任一叶节点长度为N的路径均对应一个译码序列（含固定比特）。

Decoding Algorithm for Polar Codes

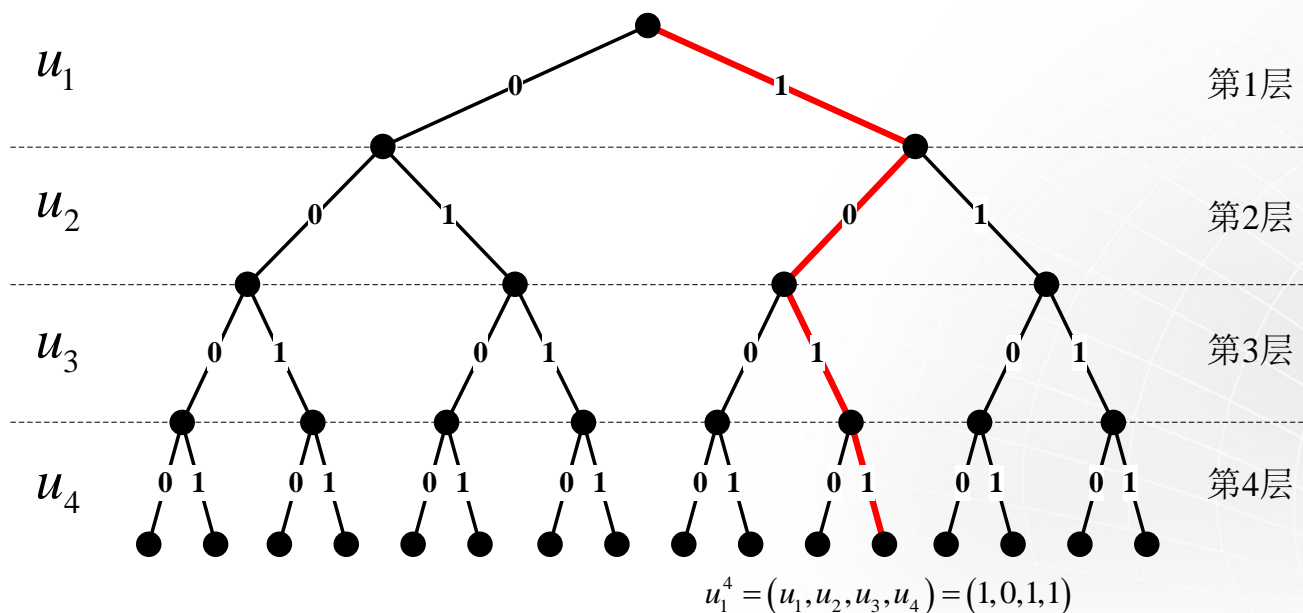
What is the TREE for polar decoding?



- 码长为N 的Polar码，都对应一棵深度为N 的满二叉树；
- 每一层边都分别对应一个信息比特或固定比特；
- 除叶节点外，每一个节点与其左、右两个后继节点之间的边分别被标记为0和1；
- 从根节点出发到任一叶节点长度为N的路径均对应一个译码序列（含固定比特）。

Decoding Algorithm for Polar Codes

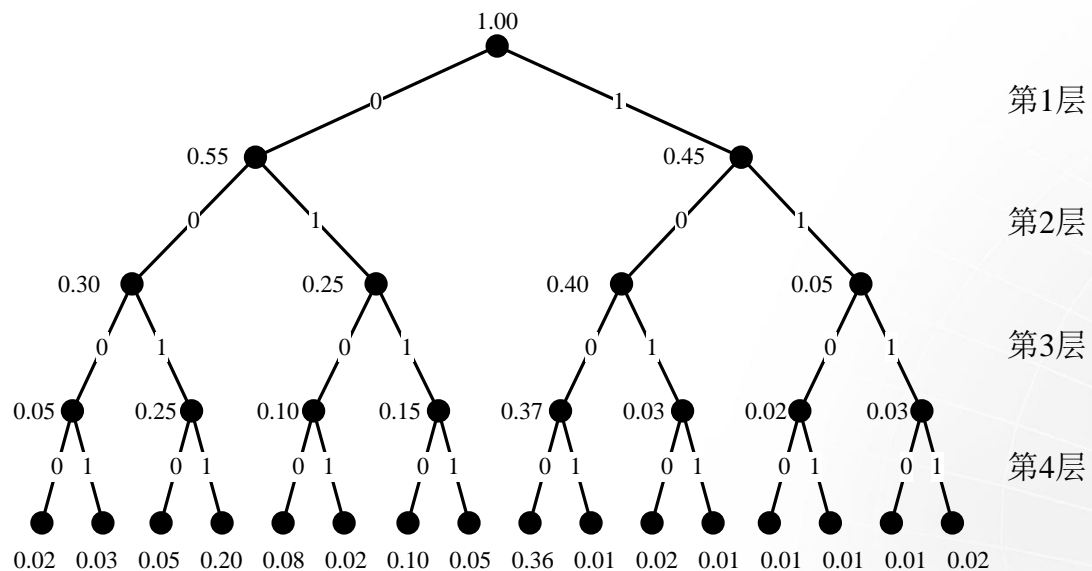
What is the TREE for polar decoding?



- 码长为N 的Polar码，都对应一棵深度为N 的满二叉树；
- 每一层边都分别对应一个信息比特或固定比特；
- 除叶节点外，每一个节点与其左、右两个后继节点之间的边分别被标记为0和1；
- 从根节点出发到任一叶节点长度为N的路径均对应一个译码序列（含固定比特）。

Decoding Algorithm for Polar Codes

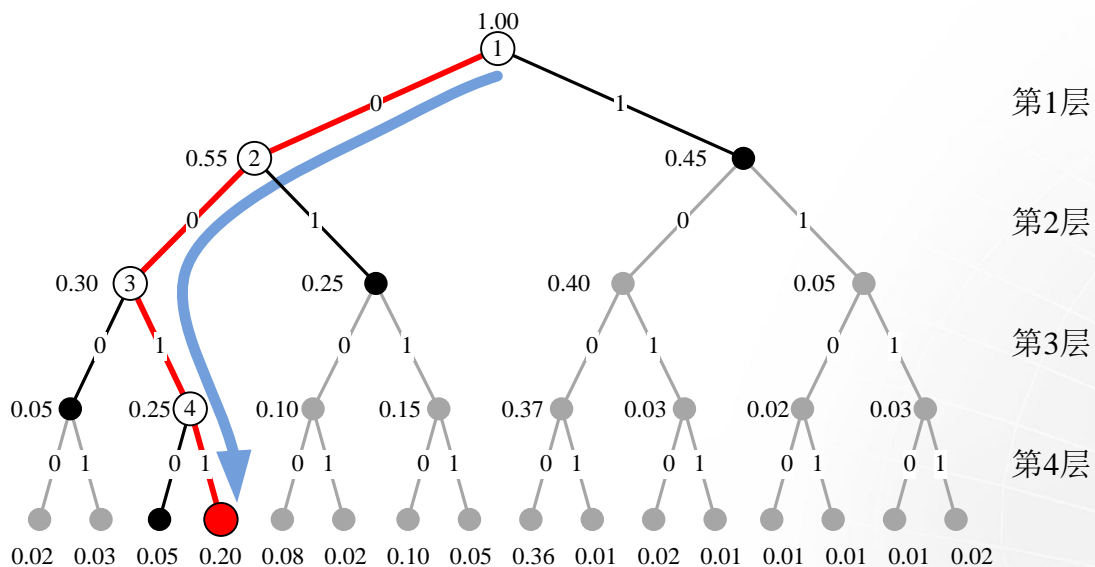
Which paths should be extended?



➤从根节点到任何一个节点所形成的路径，均对应一个路径度量值；

Decoding Algorithm for Polar Codes

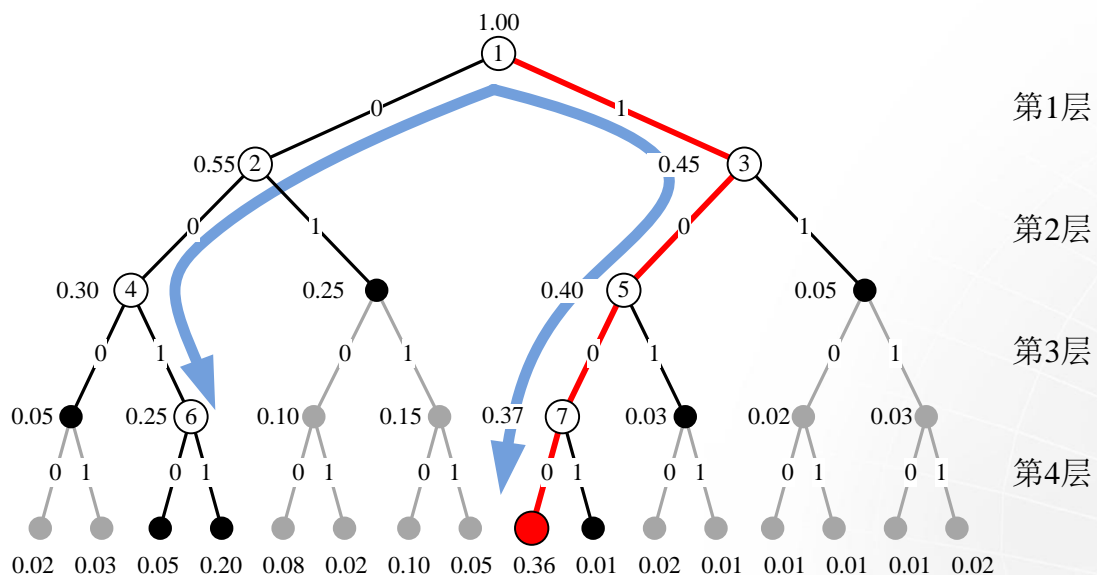
Which paths should be extended?



- 从根节点到任何一个节点所形成的路径，均对应一个路径度量值；
- 从根节点出发，按广度优先的方法对路径进行扩展；
- 每一层向下一层扩展时，选择当前层中具有**较大路径度量值的 L 条**，L 称为搜索宽度；

Decoding Algorithm for Polar Codes

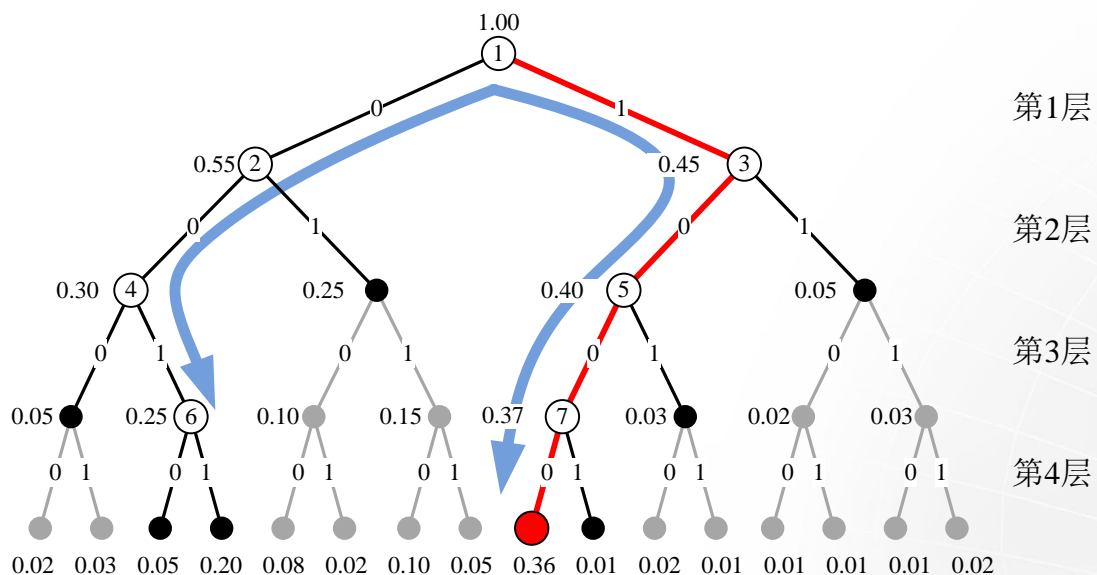
Which paths should be extended?



- 从根节点到任何一个节点所形成的路径，均对应一个路径度量值；
- 从根节点出发，按广度优先的方法对路径进行扩展；
- 每一层向下一层扩展时，选择当前层中具有**较大路径度量值的 L 条**，L 称为搜索宽度；

Decoding Algorithm for Polar Codes

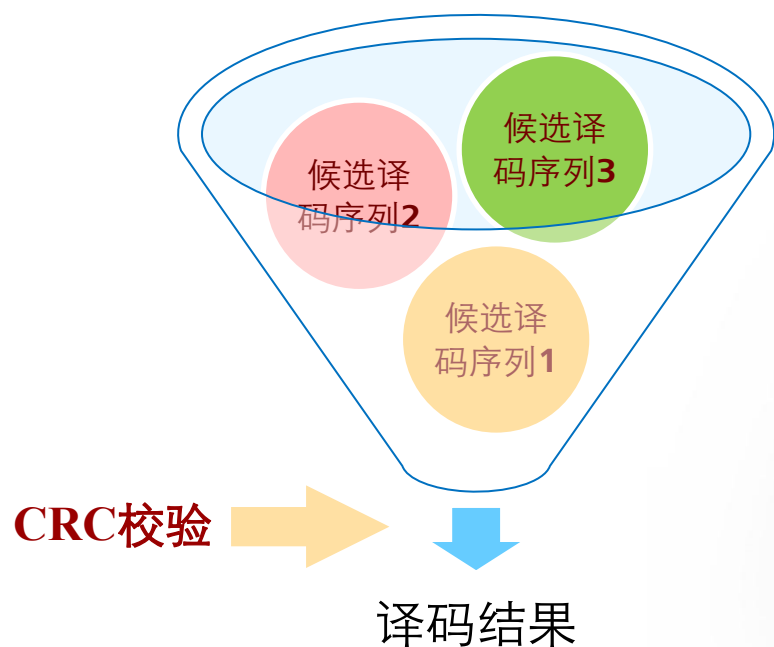
Which paths should be extended?



- 从根节点到任何一个节点所形成的路径，均对应一个路径度量值；
- 从根节点出发，按广度优先的方法对路径进行扩展；
- 每一层向下一层扩展时，选择当前层中具有**较大路径度量值的 L 条**，L 称为搜索宽度；
- 抵达叶节点层后，按度量值从大到小的顺序输出 L 条路径所对应的译码序列，作为**候选译码序列集合**；

Decoding Algorithm for Polar Codes

Which paths should be extended?



- 从根节点到任何一个节点所形成的路径，均对应一个路径度量值；
- 从根节点出发，按广度优先的方法对路径进行扩展；
- 每一层向下一层扩展时，选择当前层中具有**较大路径度量值的 L 条**，L 称为搜索宽度；
- 抵达叶节点层后，按度量值从大到小的顺序输出 L 条路径所对应的译码序列，作为**候选译码序列集合**；
- 从候选码字集合中，选出（能够通过CRC校验的）度量值最大的译码序列，提取其中的信息比特序列输出。

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

- 路径度量定义为该路径所对应的译码序列的概率，实现时往往采用其对数形式

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

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

- 路径度量定义为该路径所对应的译码序列的概率，实现时往往采用其对数形式

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

- 度量值按以下公式递归地计算得到：

$$PM(u_1^i) = \begin{cases} PM(u_1^{i-1}) & \text{若 } u_i \text{ 为信息比特或正确固定比特, 且 } (1-2u_i) = \text{sign}(L_N^{(i)}(y_1^N, u_1^{i-1})) \\ PM(u_1^{i-1}) - |L_N^{(i)}(y_1^N, u_1^{i-1})| & \text{若 } u_i \text{ 为信息比特或正确固定比特, 且 } (1-2u_i) \neq \text{sign}(L_N^{(i)}(y_1^N, u_1^{i-1})) \\ -\infty & \text{若 } u_i \text{ 为固定比特, 且取值错误} \end{cases}$$

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

- 路径度量定义为该路径所对应的译码序列的概率，实现时往往采用其对数形式

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

- 度量值按以下公式递归地计算得到：

$$PM(u_1^i) = \begin{cases} PM(u_1^{i-1}) & \text{若 } u_i \text{ 为信息比特或正确固定比特, 且 } (1-2u_i) = \text{sign}(L_N^{(i)}(y_1^N, u_1^{i-1})) \\ PM(u_1^{i-1}) - |L_N^{(i)}(y_1^N, u_1^{i-1})| & \text{若 } u_i \text{ 为信息比特或正确固定比特, 且 } (1-2u_i) \neq \text{sign}(L_N^{(i)}(y_1^N, u_1^{i-1})) \\ -\infty & \text{若 } u_i \text{ 为固定比特, 且取值错误} \end{cases}$$

其中, $PM(\phi) = 0$

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

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

其中

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



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

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

- 路径度量定义为该路径所对应的译码序列的概率，实现时往往采用其对数形式

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

- 度量值按以下公式递归地计算得到：

$$PM(u_1^i) = \begin{cases} PM(u_1^{i-1}) & \text{若 } u_i \text{ 为信息比特或正确固定比特, 且 } (1-2u_i) = \text{sign}(L_N^{(i)}(y_1^N, u_1^{i-1})) \\ PM(u_1^{i-1}) - |L_N^{(i)}(y_1^N, u_1^{i-1})| & \text{若 } u_i \text{ 为信息比特或正确固定比特, 且 } (1-2u_i) \neq \text{sign}(L_N^{(i)}(y_1^N, u_1^{i-1})) \\ -\infty & \text{若 } u_i \text{ 为固定比特, 且取值错误} \end{cases}$$

其中, $PM(\phi) = 0$

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

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

给定序列 u_1^N , 其序号为奇数与偶数的元素所构成的子序列分别用 $u_{1,o}^N$ 和 $u_{1,e}^N$ 表示, 即

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

Decoding Algorithm for Polar Codes

Summary

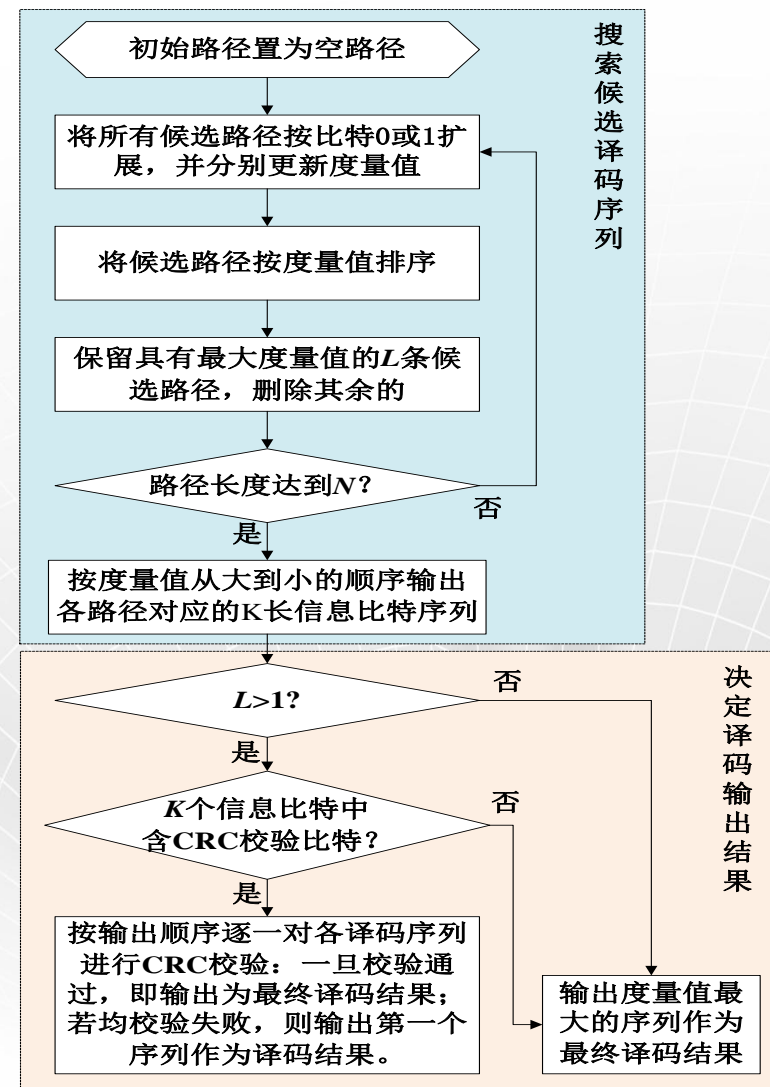
译码输入

N	码长，即编码比特序列的长度，取值为2的幂次
y_1^N	BPSK调制AWGN信道接收信号序列
A	信息比特对应的序号集合，该集合大小为 K
L	译码算法搜索宽度，自然数
σ^2	AWGN噪声方差

*该输入参数表仅适合BI-AWGN信道仿真

译码输出

u_A	长度为 K 的信息比特序列
-------	-----------------



Simple Example of Polar Encoding and Decoding

ENCODING...

➤ 输入编码器的比特序列为(1, 0)，码长设定为4，即 $K=2$ 、 $N=4$ ，码率为

$$R = \frac{K}{N} = \frac{1}{2}$$

➤ 根据Polar码构造*， u_1 与 u_2 为固定比特，取值均为0；

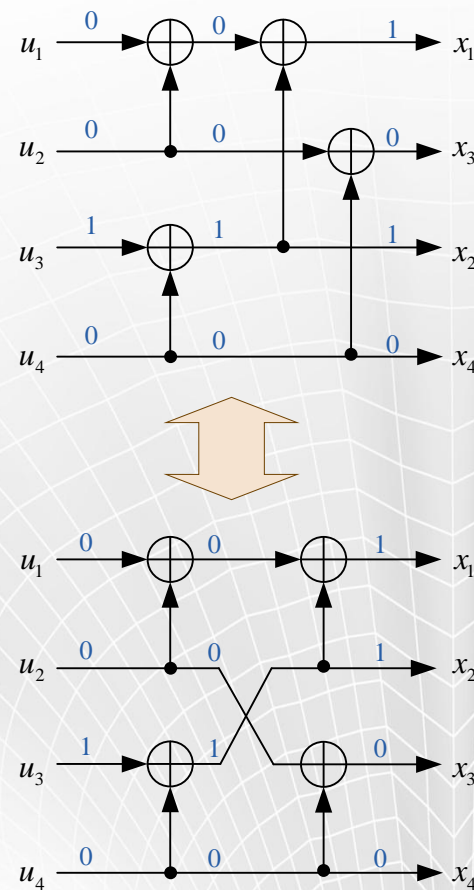
➤ 将信息比特(1,0)分别赋值给 u_3 与 u_4 ，得到

$$(u_1, u_2, u_3, u_4) = (0, 0, 1, 0)$$

➤ 利用右图编码结构，得到编码码字

$$(x_1, x_2, x_3, x_4) = (1, 1, 0, 0)$$

■ 注意到右侧上图中的编码输出比特是按照比特反序排列的；
若将编码输出比特按自然序排列，得到右侧下图中的等价结构。



Simple Example of Polar Encoding and Decoding

TRANSMIT OVER BI-AWGN CH...

- 将二进制比特序列映射为实数序列，映射关系为

$$s_i = \begin{cases} +1 & \text{若 } x_i = 0 \\ -1 & \text{若 } x_i = 1 \end{cases}$$

- 根据编码器输出得到：

$$(s_1, s_2, s_3, s_4) = (-1, -1, +1, +1)$$

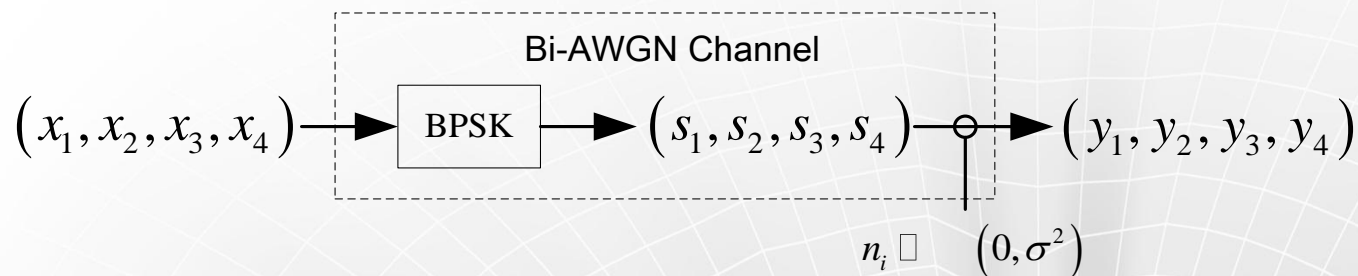
- 对上述序列加以加性高斯白噪，得到接收信号

$$y_i = s_i + n_i$$

噪声值 n_i 服从均值为0，方差为 σ^2 的正态分布，各噪声值相互独立；

- 此例中设 $\sigma^2 = 0.5$ ，噪声样值分别为 $(n_1, n_2, n_3, n_4) = (-0.1, -0.6, -1.5, 0.2)$ ；

- 那么，经过信道后得到接收序列 $(y_1, y_2, y_3, y_4) = (-1.1, -1.6, -0.5, 1.2)$ 。



Simple Example of Polar Encoding and Decoding

DECODING...

- 根据接收序列计算各个码字比特的对数似然比 (LLR)

$$z_i = L(y_i) = \ln \frac{\Pr\{x_i = 0|y_i\}}{\Pr\{x_i = 1|y_i\}} = \frac{2y_i}{\sigma^2}$$

- 得到 $(z_1, z_2, z_3, z_4) = (-4.4, -6.4, -2.0, 4.8)$

- 本例中，设定搜索宽度 $L=1$ ，即每一层仅保留1条路径用作下一层的扩展；

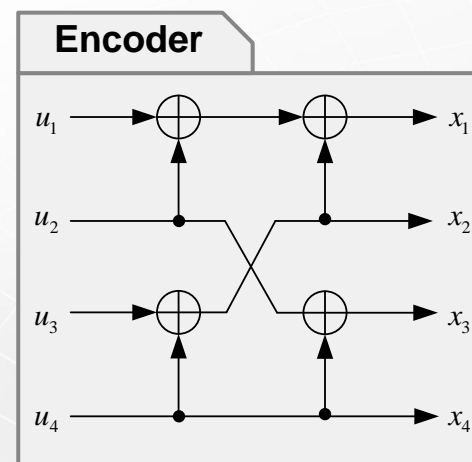
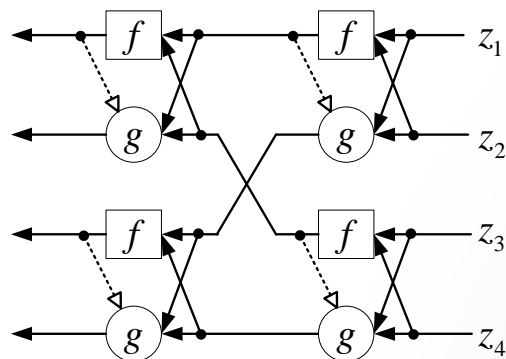
- 注意到当 $L=1$ 时，SCL译码算法退化为了SC译码算法。

- 初始化 $L(y_i) = z_i$ ，候选路径表仅包含一条空路径，该路径的度量值为 $PM(\phi) = \ln(1.0) = 0$ 。

Simple Example of Polar Encoding and Decoding

DECODING...

➤ 在译码时可以用下图结构进行路径度量值的计算：

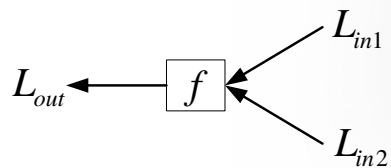
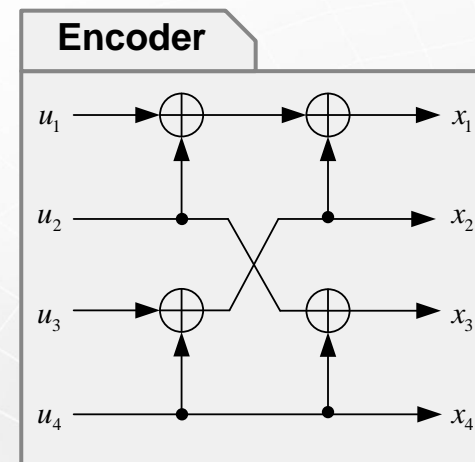
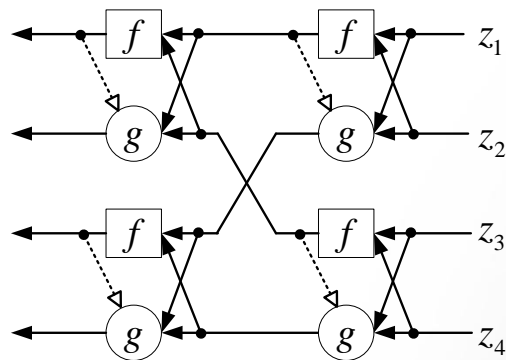


- 将编码结构中的模二加 \oplus 和链接点 \cdot 分别用 f 和 g 替换；
- 反向信号流方向：编码时从 u 到 x ，译码时候从 z （对应 x ）到 u 。

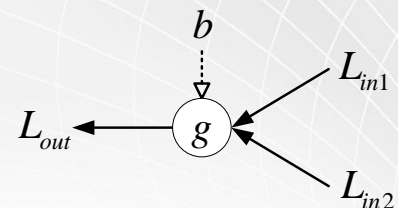
Simple Example of Polar Encoding and Decoding

DECODING...

➤ 在译码时可以用下图结构进行路径度量值的计算：



$$L_{out} = f(L_{in1}, L_{in2}) = \text{sign}(L_{in1} \cdot L_{in2}) \cdot \min(|L_{in1}|, |L_{in2}|)$$



$$L_{out} = g(L_{in1}, L_{in2}, b) = (-1)^b \cdot L_{in1} + L_{in2}$$

Simple Example of Polar Encoding and Decoding

DECODING...

- 计算第一个比特相关的LLR:

$$L_4^{(1)}(y_1^4) = f(L_2^{(1)}(y_1^2), L_2^{(1)}(y_3^4))$$

其中, $L_2^{(1)}(y_1^2)$ 和 $L_2^{(1)}(y_3^4)$ 递归地计算为

$$L_2^{(1)}(y_1^2) = f(L_1^{(1)}(y_1), L_1^{(1)}(y_2)) = f(z_1, z_2) = f(-4.4, -6.4) = 4.4$$

$$L_2^{(1)}(y_3^4) = f(L_1^{(1)}(y_3), L_1^{(1)}(y_4)) = f(z_3, z_4) = f(-2.0, 4.8) = -2.0$$

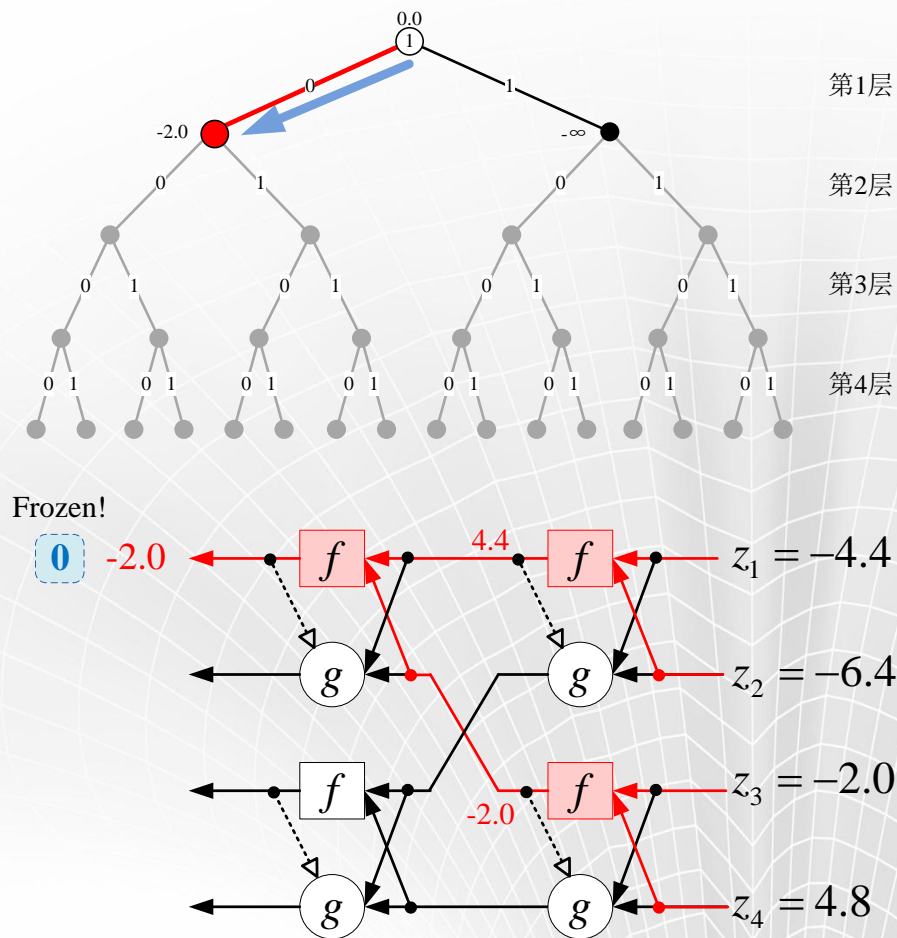
从而, $L_4^{(1)}(y_1^4) = f(4.4, -2.0) = -2.0$

- 计算候选路径的度量值

$$PM(u_1 = 0) = PM(\phi) - |L(y_1^4)| = 0 - |-2.0| = -2.0$$

$$PM(u_1 = 1) = -\infty$$

- 由于 $L=1$, 仅选择 $u_1 = 0$ 用作后续的路径扩展。



Simple Example of Polar Encoding and Decoding

DECODING...

- 计算第二个比特相关的LLR:

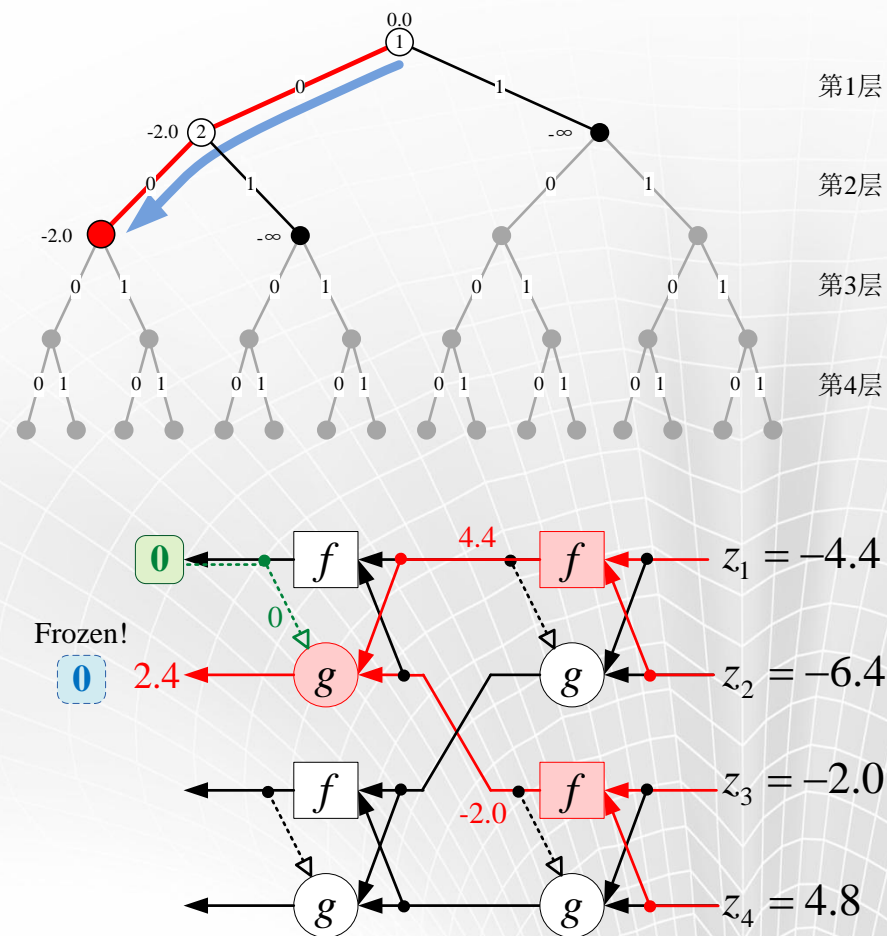
$$\begin{aligned} L_4^{(2)}(y_1^4, u_1) &= g(L_2^{(1)}(y_1^2), L_2^{(1)}(y_3^4), u_1) \\ &= g(4.4, -2.0, 0) \\ &= 2.4 \end{aligned}$$

- 计算候选路径的度量值

$$PM(u_1^2 = 00) = PM(u_1 = 0) = -2.0$$

$$PM(u_1^2 = 01) = -\infty$$

- 选择具有较大度量值的 $u_1^2 = (00)$ 用作后续的路径扩展。



Simple Example of Polar Encoding and Decoding

DECODING...

- 计算第三个比特相关的LLR:

$$L_4^{(3)}(y_1^4, u_1^2) = f(L_2^{(2)}(y_1^2, u_1 \oplus u_2), L_2^{(2)}(y_3^4, u_2))$$

其中, $L_2^{(2)}(y_1^2, u_1 \oplus u_2)$ 和 $L_2^{(2)}(y_3^4, u_2)$ 递归地计算为

$$L_2^{(2)}(y_1^2, u_1 \oplus u_2) = g(L_1^{(1)}(y_1), L_1^{(1)}(y_2), u_1 \oplus u_2) = g(-4.4, -6.4, 0) = -10.8$$

$$L_2^{(2)}(y_3^4, u_2) = g(L_1^{(1)}(y_3), L_1^{(1)}(y_4), u_2) = g(-2.0, 4.8, 0) = 2.8$$

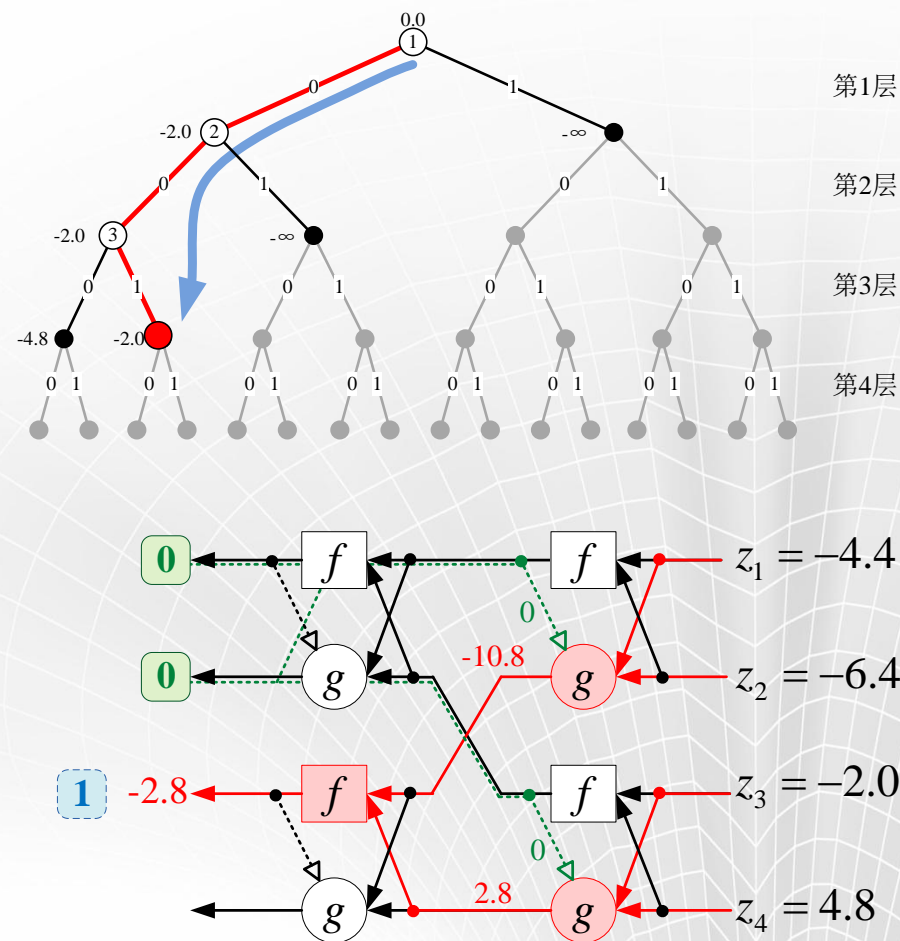
从而, $L_4^{(3)}(y_1^4, u_1^2) = f(-10.8, 2.8) = -2.8$;

- 计算候选路径的度量值

$$PM(u_1^3 = 000) = PM(u_1^2 = 00) - |L(y_1^4, u_1^2)| = -2.0 - 2.8 = -4.8$$

$$PM(u_1^3 = 001) = PM(u_1^2 = 00) = -2.0$$

- 选择具有较大度量值的 $u_1^3 = (001)$ 用作后续的路径扩展。



Simple Example of Polar Encoding and Decoding

DECODING...

- 计算第四个比特相关的LLR

$$\begin{aligned} L_4^{(4)}(y_1^4, u_1^3) &= g(L_2^{(2)}(y_1^2, u_1 \oplus u_2), L_2^{(2)}(y_3^4, u_2), u_3) \\ &= g(-10.8, 2.8, 1) = 13.6 \end{aligned}$$

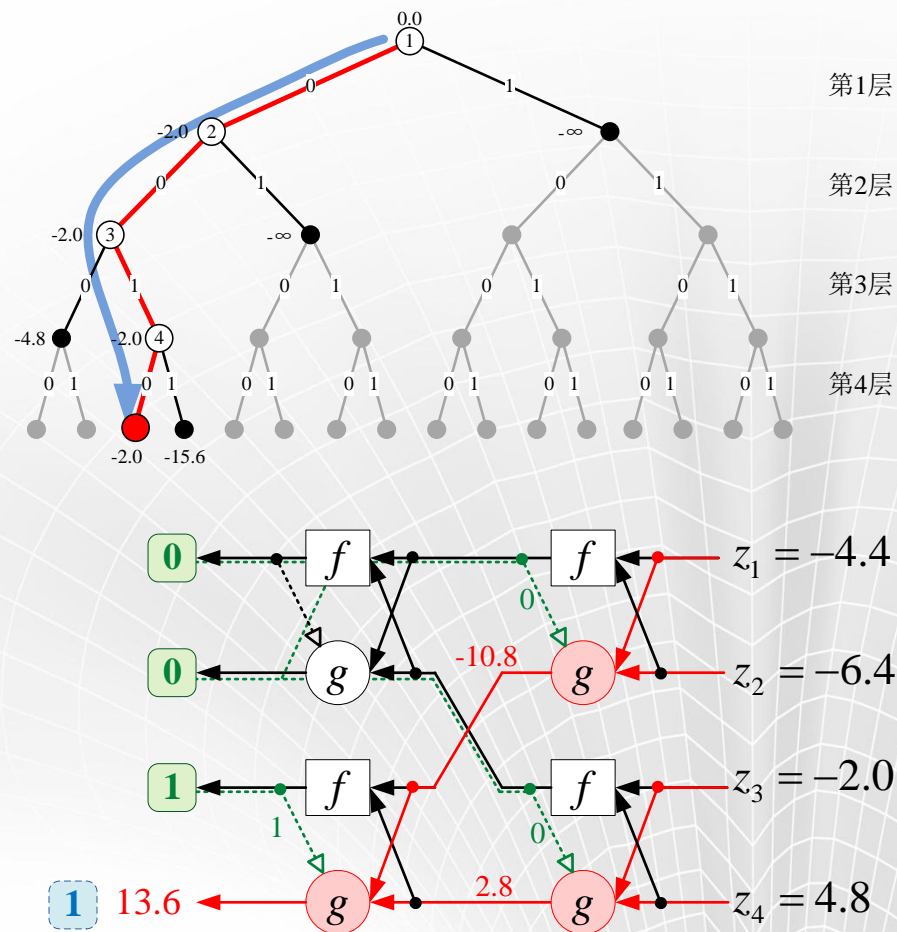
- 计算候选路径的度量值

$$PM(u_1^4 = 0010) = PM(u_1^3 = 001) = -2.0$$

$$PM(u_1^4 = 0011) = PM(u_1^3 = 001) - |L(y_1^4, u_1^3)| = -2.0 - 13.6 = -15.6$$

- 搜索已经达到叶子节点，因此选择具有较大路径度量值的 $u_1^4 = (0010)$ ，并输出为候选译码序列。

- 由于 $L=1$ ， $u_1^4 = (0010)$ 是唯一的候选序列，因此最终译码器根据该序列抽取译码信息比特为 (10) ，即 $u_A = (10)$ ；对比发送序列可知，此次译码结果正确。



How polar code will be implemented for this task

IMPLEMENTATION GUIDE

Phase I

■ 设计目标:

- 正确理解Polar码编译码算法，编写Polar码编译码算法仿真程序（C或Matlab），并输出正确的仿真结果；
- 设计Polar码编译码器FPGA结构，并评估相应的FPGA资源、数据处理速率、时延等

■ 交付件:

- Polar码算法程序说明文档：BPSK调制、AWGN信道下，Polar码BER与BLER性能仿真结果；
- Polar码算法仿真程序；
- Polar码FPGA概要设计文档。

■ 评价标准:

- Polar码算法性能仿真结果通过课题组所提供的对比性能曲线，输出正确的编译码程序及仿真结果；
- Polar码FPGA设计架构的优劣，例如能够支撑的算法搜索路径宽度L的大小、占用的FPGA资源、数据处理速度、数据处理时延等

Phase II

■ 设计目标:

- 根据Polar码编译码算法设计正确的FPGA架构，并用硬件描述语言（Verilog RTL等）正确实现Polar码编译码算法；
- 利用本赛所提供的FPGA硬件，进行对所设计的FPGA链路进行测试。

■ 交付件:

- Polar码FPGA设计说明文档；
- Polar码FPGA设计程序及FPGA加载文件；
- Polar码FPGA测试结果。

■ 评价标准:

- Polar码FPGA设计、实现及FPGA测试的功能正确性
- FPGA设计能够支撑的算法搜索路径宽度L的大小、占用的FPGA资源、数据处理速度、数据处理时延等

System Configuration Parameters for Implementation

参数	值	描述
N	256或1024	Polar码的码长（编码器输出序列长度）为256或1024比特； Matlab/C仿真程序要求支持全部两种码长； FPGA设计时，可以根据能力及FPGA资源选择一种或两种码长。
K	128或512	信息序列长度（编码器输入序列长度）：N=256时，K=128；N=1024时，K=512；其中包含有24个CRC比特。。
$g(D)$	根据LTE协议	CRC生成多项式， $g(D)=D^{24}+D^{23}+D^{18}+D^{17}+D^{14}+D^{11}+D^{10}+D^7+D^6+D^5+D^4+D^3+D+1$
L	1~32	译码搜索宽度，一般取2的幂次，如1、2、4、8、16、32，实现时可以根据能力及FPGA资源估计选择合适的值（越大性能越好，但复杂度也越高）
A	取下页序列中1的位置	信息比特位置集合，对编、译码器均已知，大小为K
u_{A^c}	全零	固定比特序列，长度为(N-K)=128或512，对编、译码器均已知

[illegible]

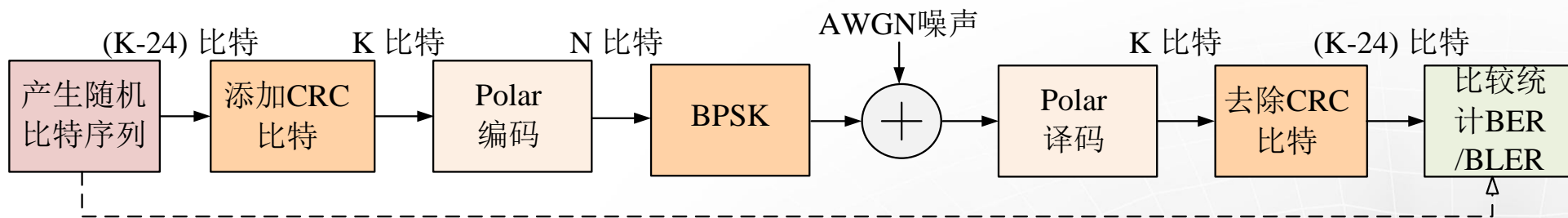
- Page 48

System Configuration Parameters for Implementation

[illegible]

- 本次大赛所用信息/固定比特位置配置（按行读），“0”表示固定比特位置，“1”表示信息比特位置
- $N=1024$, $K=512$

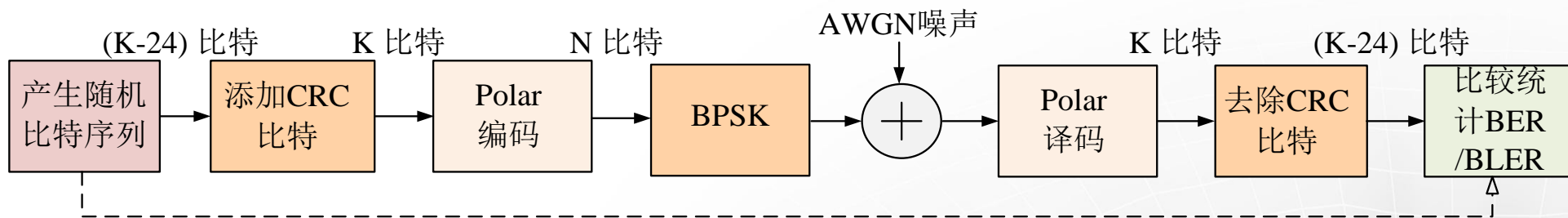
System Diagram for Simulation and FPGA



Polar码算法性能仿真链路框图

- 源序列为长度为 $(K-24)$ 比特的二进制序列，0、1等概率分布；
- 送入Polar编码器的信息比特序列包括有24个CRC比特；
- 解调和译码合并入Polar译码模块；
- 将发送的 $(K-24)$ 的源序列直接与译码、去CRC后得到的结果进行比较，并统计BER/BLER
 - BER（误比特率）为总的错误比特数量除于总的发送比特数量；
 - BLER（误码块率）为总的错误译码序列数量（任一比特错误，即判定为错误）除于总的发送序列数量。

System Diagram for Simulation and FPGA

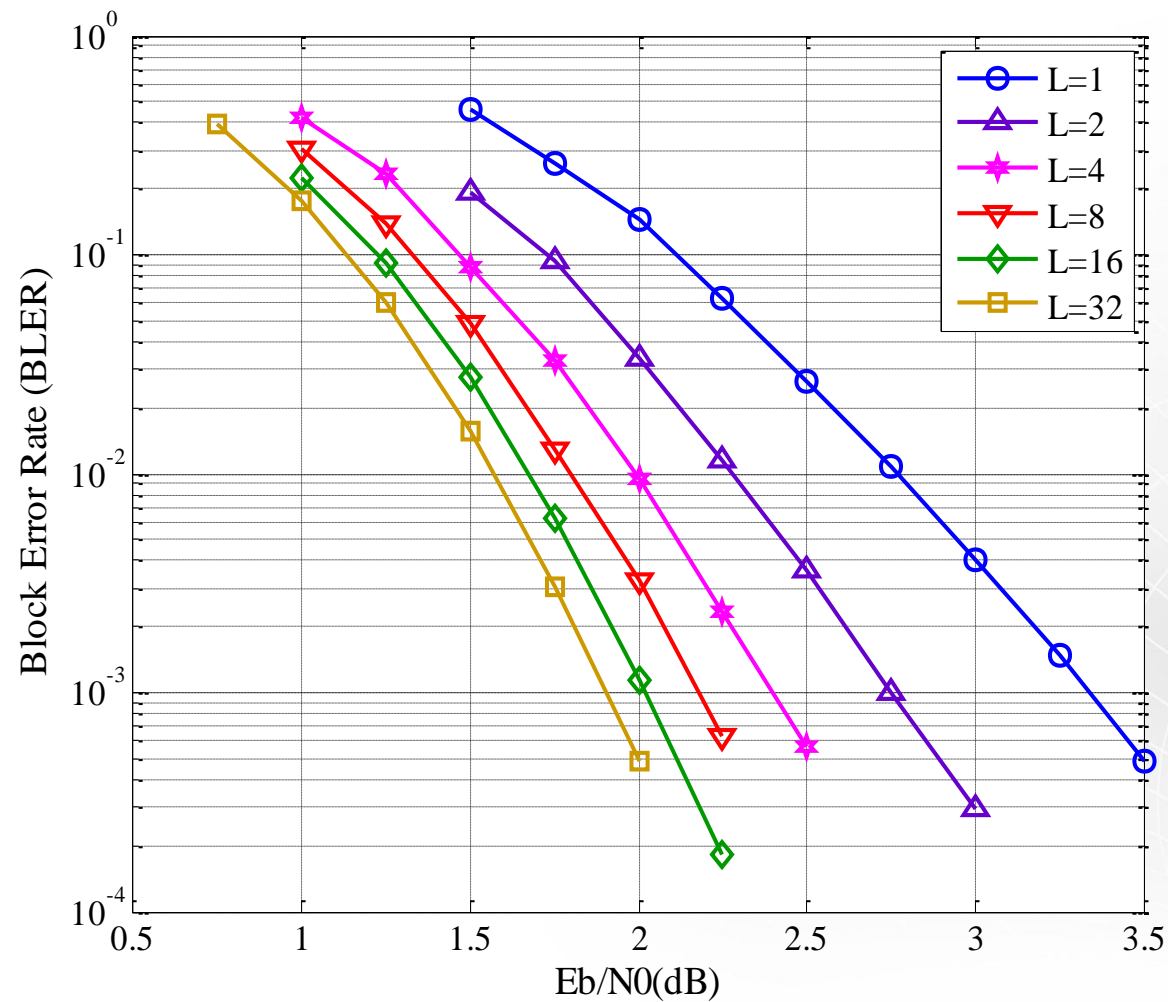


Polar码算法性能仿真链路框图

- BPSK将比特0或者比特1映射成实数信号+1或者-1: $\begin{cases} 0 \rightarrow +1 \\ 1 \rightarrow -1 \end{cases}$
- AWGN噪声服从均值为0、方差为 σ^2 的高斯分布
 - 长度为 N 、方差为 $\sigma^2 = 0.64$ 的噪声序列用Matlab代码生成为: `noises = sqrt(0.64)*randn(1, N);`
 - 噪声方差根据仿真目标比特信噪比 (E_b/N_0) 计算得到, 其关系为

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

Performance Curves for Reference



- BPSK+AWGN
- $N=1024, K=512$
- CA-SCL译码
- 浮点译码器
- 仿真码块数 $> 10^5$

Hints on the Polar Decoder Implementation

■ 路径存储结构

- 直接实现译码器，存储每一条路径所需的空间复杂度为 $O(N \log N)$
- 利用SC/SCL译码的特点，实际上存储一条路径需要的空间仅为 $O(N)$

■ 路径管理

- SCL时需要大量的路径复制
- 直接进行复制，计算复杂度为 $O(N^2)$
- 路径内存共享，仅复制需要更新的部分，即所谓“Lazy Copy”，极大降低复杂度

■ 定点方案

- 定点方案的选择直接影响FPGA译码器的性能
- 建议LLR值用不小于8比特表示，而路径度量值一般不小于10比特，其中用2比特表示小数部分；
- 路径度量的取值范围为0或者负数，符号位是没有意义的。

[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

主办单位 |



西安电子科技大学
XIDIAN UNIVERSITY

terasic Terasic Inc.

赞助厂商 |



SPREADTRUM