第一届5G算法创新大赛
——Polar Code

现场宣讲&答疑会

2015年7月16日

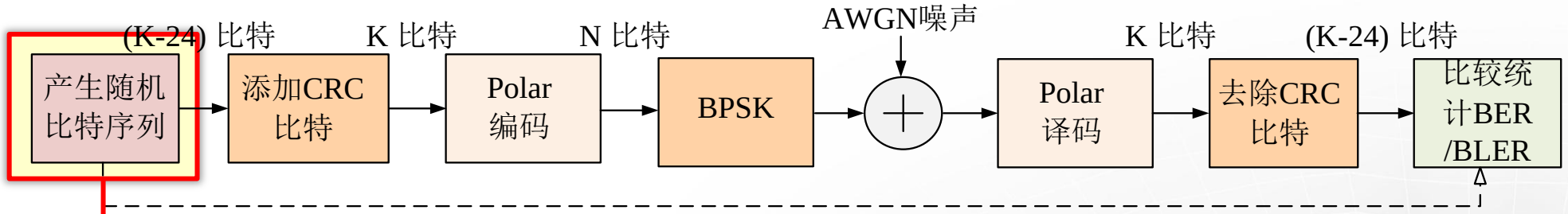
Outline

- The simulation link
- Polar encoding
- Polar decoding
- A software demo

Brief introduction of every modules on the link

THE SIMULATION LINK

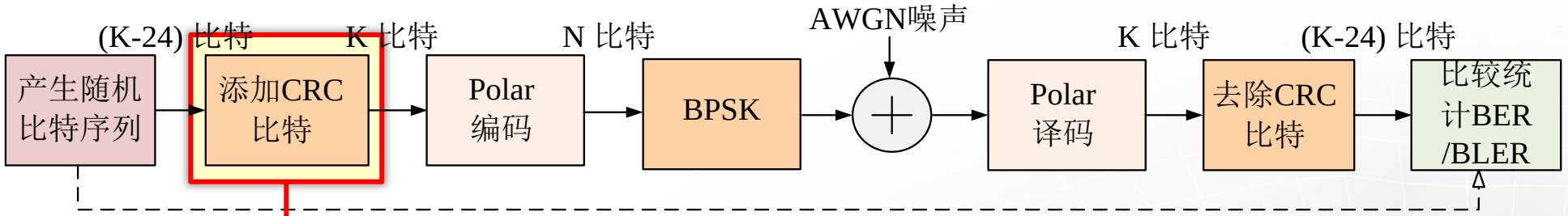
System Diagram for Simulation and FPGA



Polar码算法性能仿真链路框图

- x = genSrc(len)
  - 功能：生成0、1等概率的二进制序列。
  - 输入：len - 序列长度
  - 输出：x - 长度为len的二进制序列

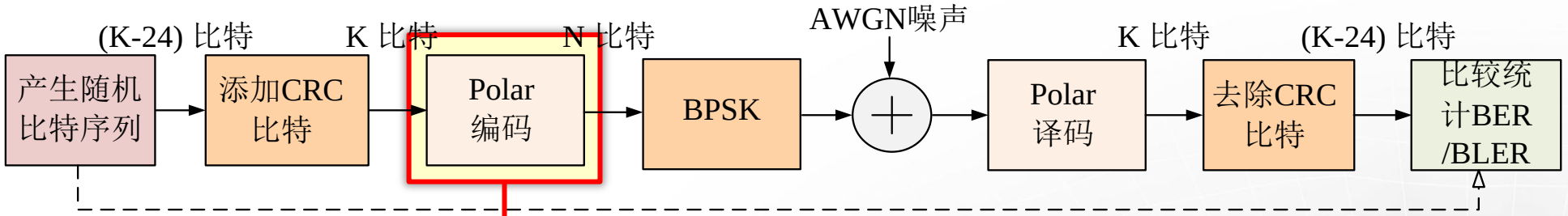
System Diagram for Simulation and FPGA



Polar码算法性能仿真链路框图

- y = attachCRC24(x, len, g)
  - 功能：添加24个CRC比特
  - 输入：
    - x - 原比特序列
    - len - x序列的长度
    - g - CRC生成多项式
  - 输出：y - 添加CRC比特后的序列，长度为len+24

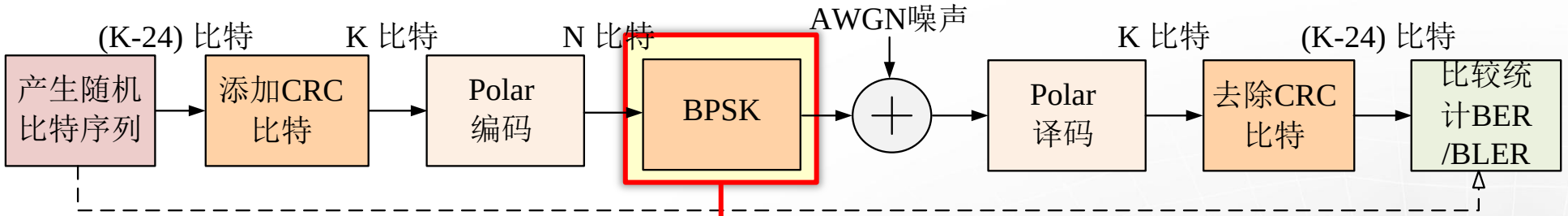
System Diagram for Simulation and FPGA



Polar码算法性能仿真链路框图

- y = polarEnc(x, K, N, infoInd)
  - 功能：Polar码编码
  - 输入：
    - x - 信息比特序列
    - K - 信息比特序列长度
    - N - 编码比特序列长度，码长
    - F - 指示固定比特位置
  - 输出：y - 编码比特序列

System Diagram for Simulation and FPGA



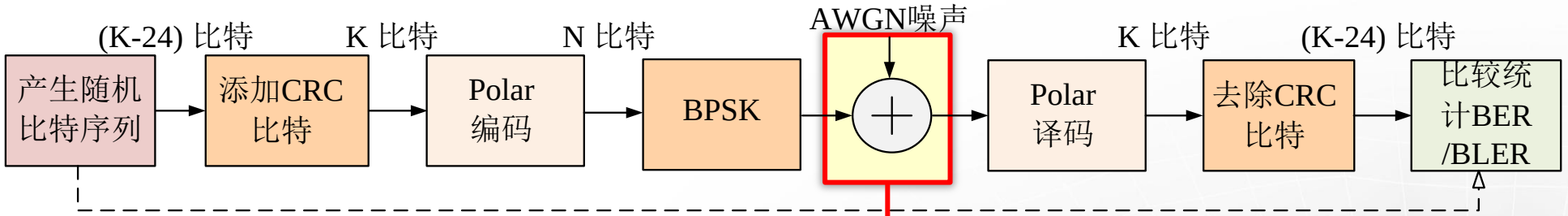
Polar码算法性能仿真链路框图

BPSK符号映射规则

$$\begin{cases} 0 \rightarrow +1 \\ 1 \rightarrow -1 \end{cases}$$

- y = bpsk(x, len)
- 功能：BPSK符号映射。
- 输入：
  - x - 发送比特序列
  - len - 发送比特序列的长度
- 输出：y - 调制符号序列

System Diagram for Simulation and FPGA



Polar码算法性能仿真链路框图

- y = awgn(x, len, N0)
  - 功能：加噪，模拟AWGN信道。
  - 输入：
    - x - 发送符号序列
    - len - 符号序列的长度
    - N0 - 单边功率谱密度
  - 输出：y - 接收符号序列

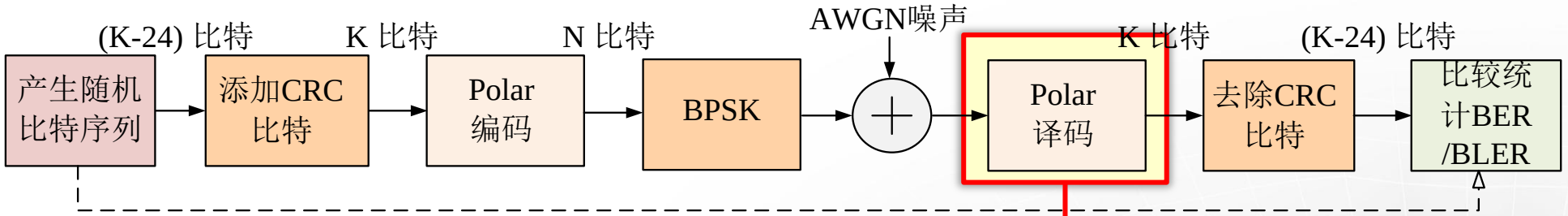
Matlab代码案例:

```
noises = sqrt(N0/2)*randn(1, N);
```

信噪比（Eb/N0）计算:

$$E_b / N_0\,(\mathrm{dB}) = 10 \times \log_{10} \frac{NE_s}{(K - L_{CRC})N_0}$$

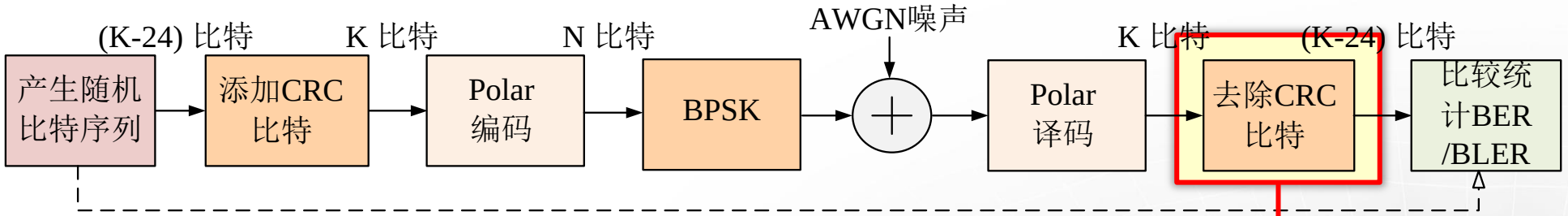
System Diagram for Simulation and FPGA



Polar码算法性能仿真链路框图

- x = polarDec(y, N, K, L, N0, F)
  - 功能：polar译码，采用SCL/CASCL算法
  - 输入：
    - y - 接收符号序列
    - N - 码长
    - K - 信息序列长度
    - L - 列表大小
    - N0 - 单边功率谱密度
    - F - 指示固定比特位置
  - 输出：y - 译码序列，长度为K

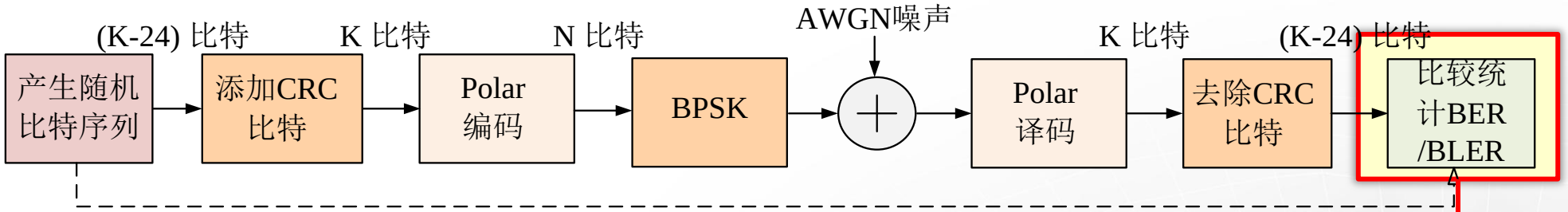
System Diagram for Simulation and FPGA



Polar码算法性能仿真链路框图

- y = detachCRC24(x, len)
  - 功能：去除24个CRC比特
  - 输入：
    - x - 含有CRC的比特序列
    - len - 输入序列长度
  - 输出：y - 去除CRC后的比特序列，长度为len-24

System Diagram for Simulation and FPGA



Polar码算法性能仿真链路框图

- e = errChk(x, y, len)
  - 功能：逐比特比较两个序列，返回不同元素的数量
  - 输入：
    - x - 源比特序列
    - y - 译码比特序列
    - len - 序列长度
  - 输出：e - 序列x与y对应位置不同的比特数。

Detailed polar encoding procedure

POLAR ENCODING

Polar Encoding

- 混合信息比特与固定比特得到 $u_1^N=(u_1,u_2,\cdots,u_N)$
  - 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。

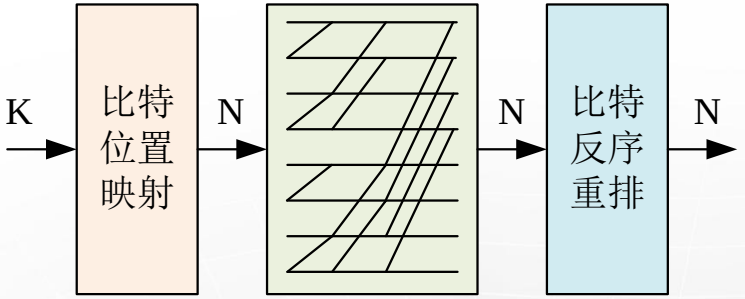
- 比特反序重排

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$(b_1,b_2,\cdots,b_n)$，

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

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



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

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

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

Polar Encoding

示例：Polar码编码 N=8 K=4

信息比特序号集合$\mathrm{A}=\{4,6,7,8\}$

固定比特序号集合$\mathrm{A}^{c}=\{1,2,3,5\}$

信息比特集合为(i_1,i_2,i_3,i_4)

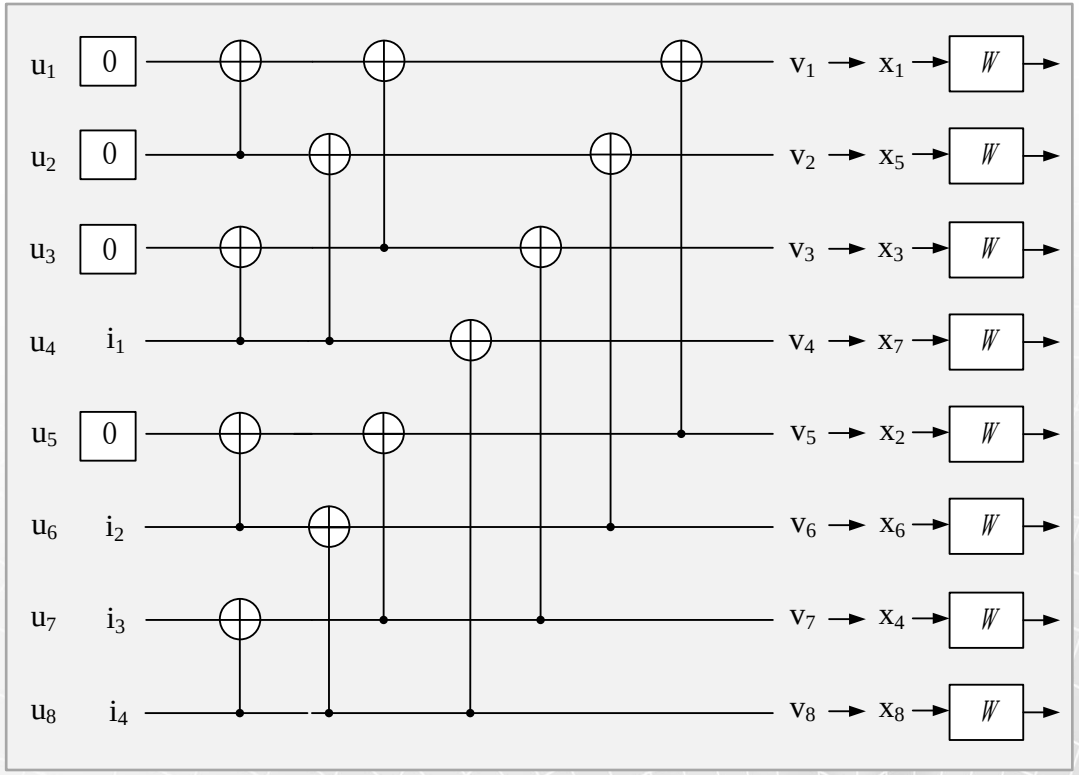
固定比特集合为$(0,0,0,0)$

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

$v_1^8=u_1^8F^{\otimes 3}$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



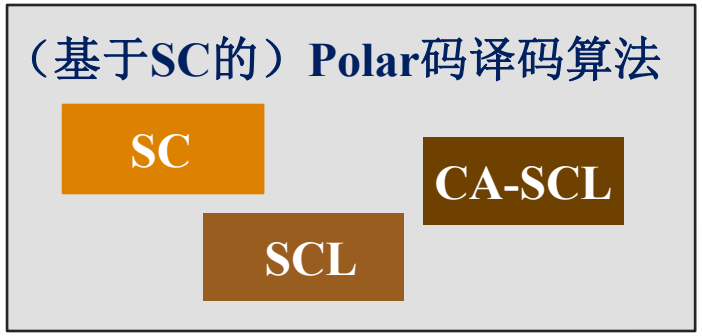
Algorithm, structure and advices

POLAR DECODING

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

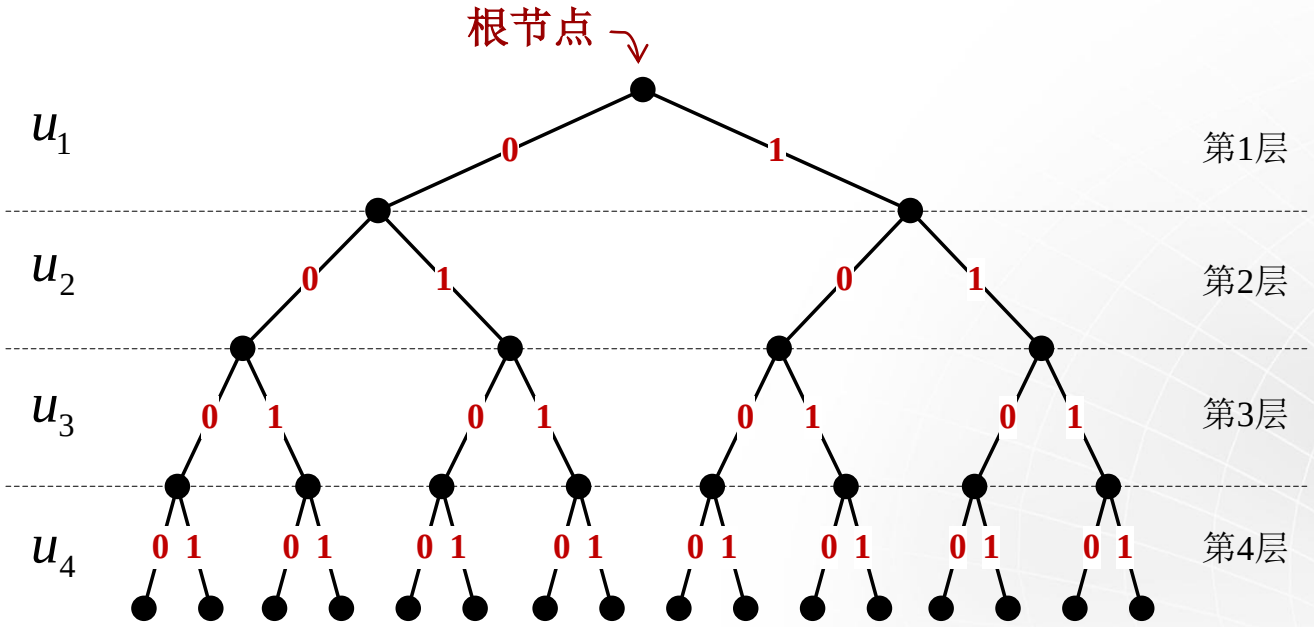
What is the TREE for polar decoding? 译码树的定义

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

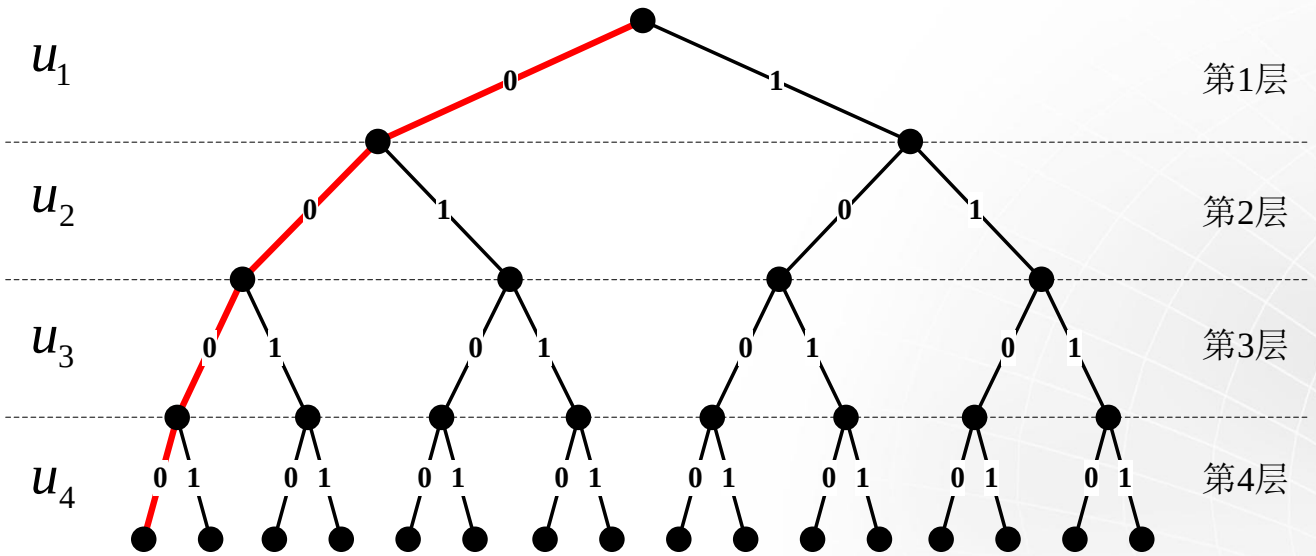
What is the TREE for polar decoding?



- 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?

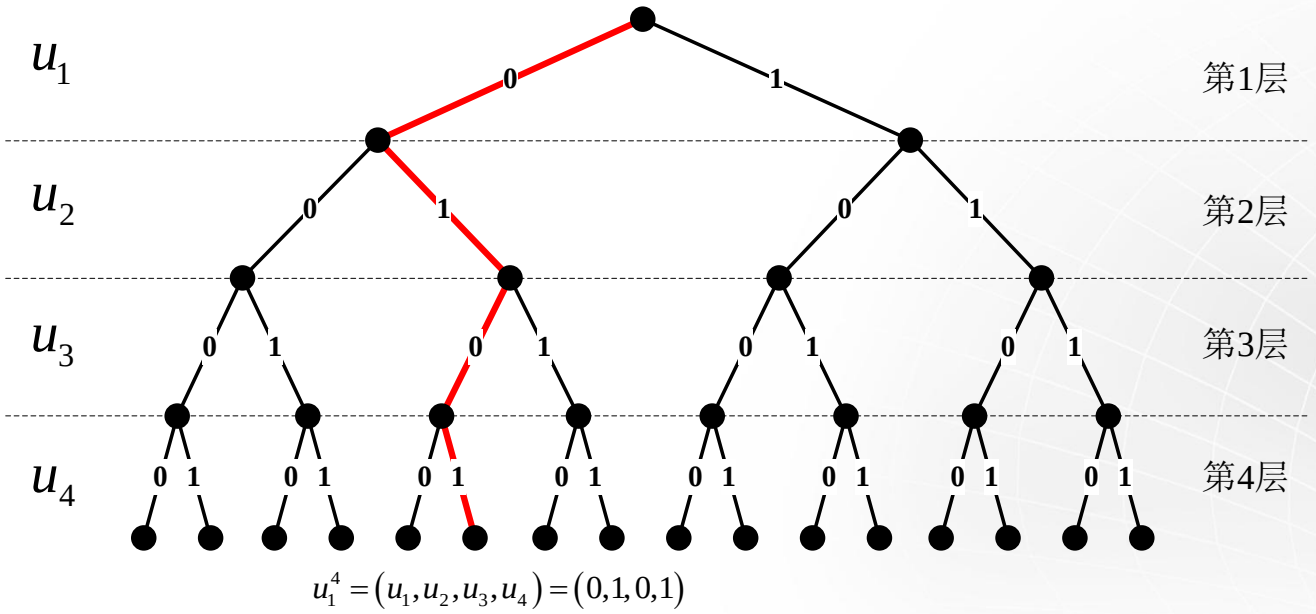


$u_1^4 = (u_1, u_2, u_3, u_4) = (0,0,0,0)$

- 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

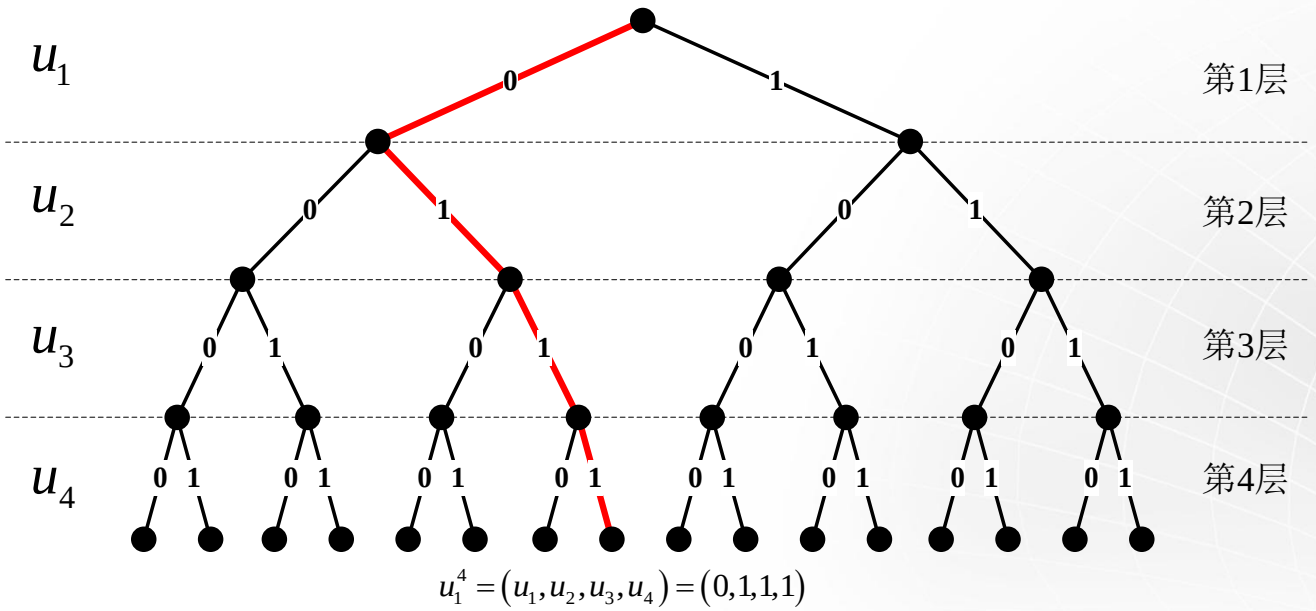
What is the TREE for polar decoding?



- 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

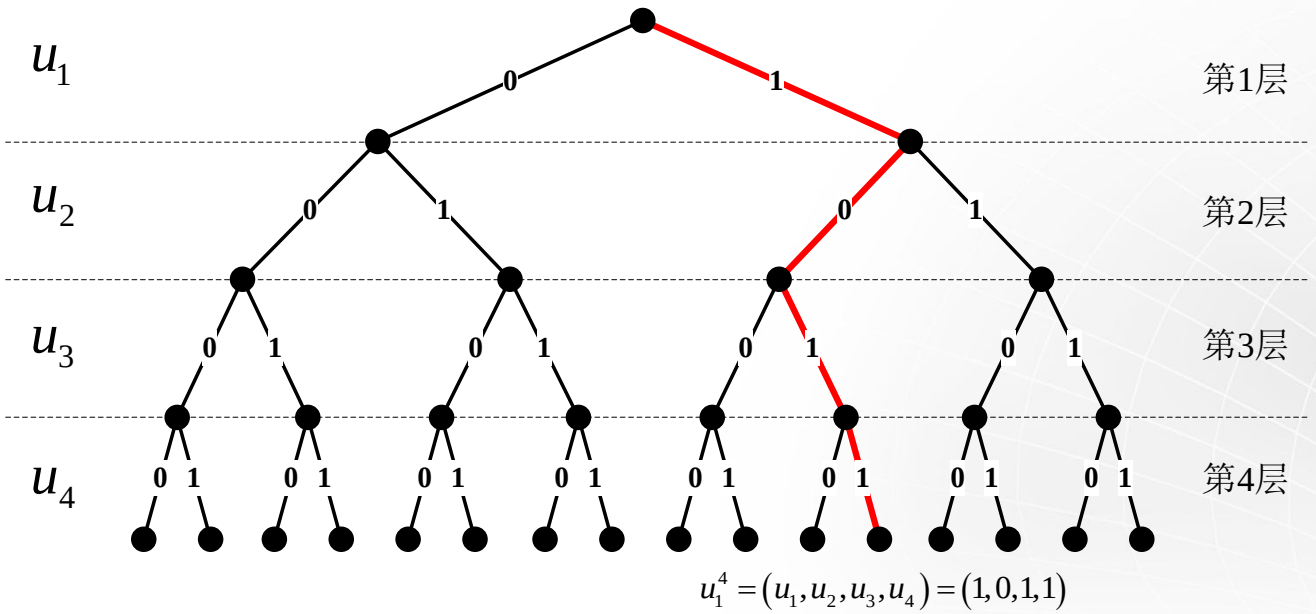
What is the TREE for polar decoding?



- 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

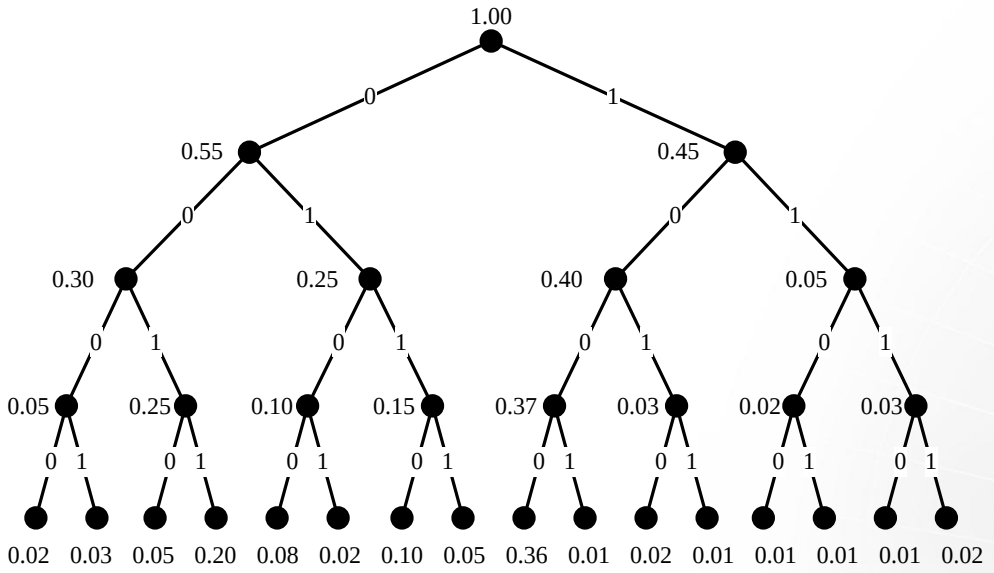
What is the TREE for polar decoding?



- 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

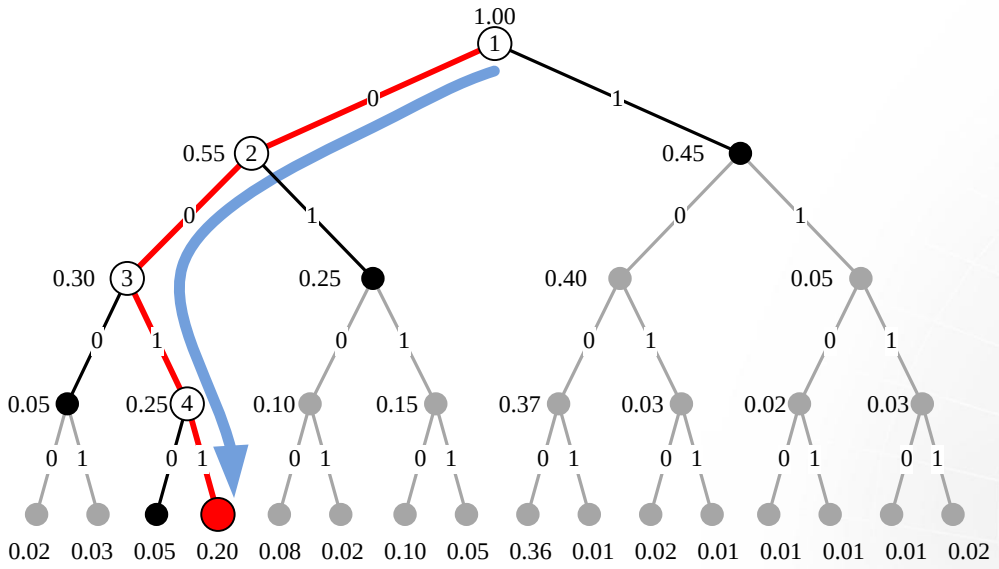
Which paths should be extended?



➢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

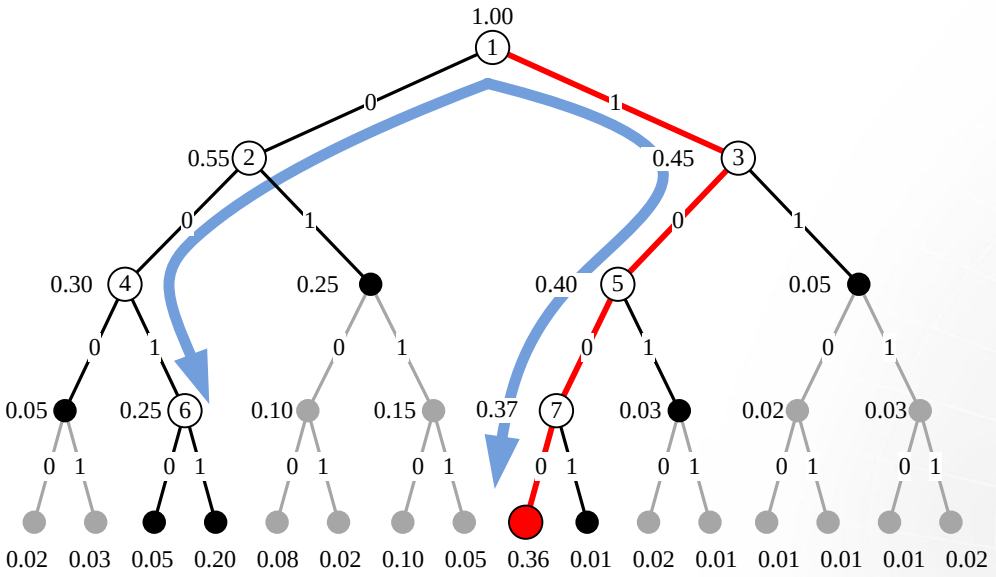
Which paths should be extended?



- 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

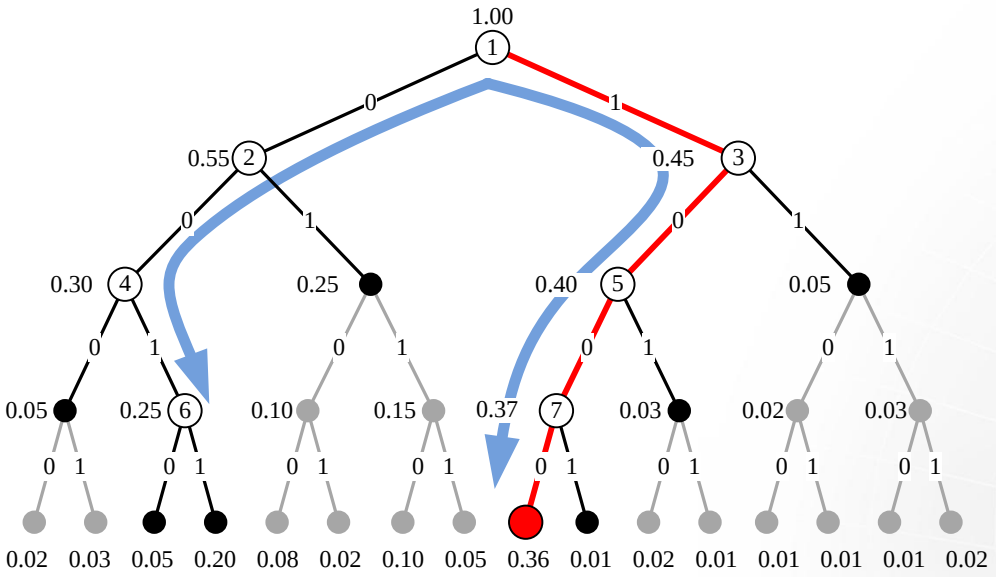
Which paths should be extended?



- 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

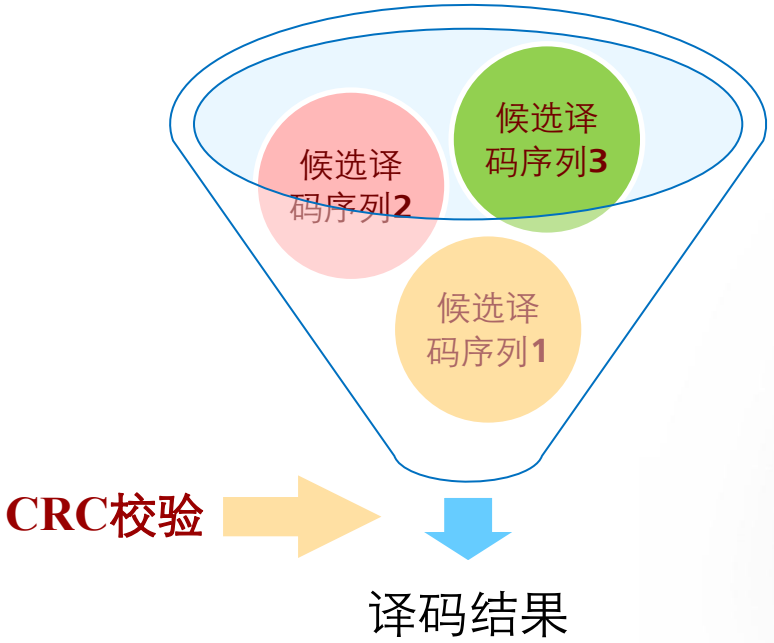
Which paths should be extended?



- 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

$$PM\left(u_1^i\right)=\ln\left(\Pr\left\{u_1^i \mid y_1^N\right\}\right)$$

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

$$PM\left(u_1^i\right)=\ln\left(\Pr\left\{u_1^i \mid y_1^N\right\}\right)$$

- 度量值按以下公式递归地计算得到：

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

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

$$PM\left(u_1^i\right)=\ln\left(\Pr\left\{u_1^i \mid y_1^N\right\}\right)$$

- 度量值按以下公式递归地计算得到：

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

其中，$PM\left(\phi\right)=0$

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

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

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

⇦ $L_1^{(1)}\left(y_i\right)=\ln\dfrac{\Pr\left\{x_i=0\middle|y_i\right\}}{\Pr\left\{x_i=1\middle|y_i\right\}}$

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

$$PM\left(u_1^i\right)=\ln\left(\Pr\left\{u_1^i \mid y_1^N\right\}\right)$$

- 度量值按以下公式递归地计算得到：

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

其中，$PM\left(\phi\right)=0$

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

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

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

⇦ $L_1^{(1)}\left(y_i\right)=\ln\dfrac{\Pr\left\{x_i=0\middle|y_i\right\}}{\Pr\left\{x_i=1\middle|y_i\right\}}$

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

$$u_{1,o}^N=\left(u_1,u_3,\ldots,u_{2k-1},\ldots,u_{N-1}\right)\qquad u_{1,e}^N=\left(u_2,u_4,\ldots,u_{2k},\ldots,u_N\right)$$

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

- 似然比递归计算详解:
  - ✓ 递归公式:

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

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

其中

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

$$u_1^{2N}=\left(u_1,u_2,\cdots,u_{2N}\right)$$

$$u_{1,o}^{2N}=\left(u_1,u_3,\cdots,u_{2i-1},\cdots,u_{2N-1}\right)$$

$$u_{1,e}^{2N}=\left(u_2,u_4,\cdots,u_{2i},\cdots,u_{2N}\right)$$

$$u_{1,o}^{2N}\oplus u_{1,e}^{2N}=\left(u_1+u_2,u_3+u_4,\cdots,u_{2i-1}+u_{2i},\cdots,u_{2N-1}+u_{2N}\right)$$

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

- 似然比递归计算详解:
  - ✓ 递归公式:

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

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

其中

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

定义 $f\left(N,i,y_1^N,u_1^{i-1}\right)=L_N^{(i)}\left(y_1^N,\hat{u}_1^{i-1}\right)$

$$f\left(N,i,y_1^N,u_1^{i-1}\right)\rightarrow f\left(\frac{N}{2},\left\lceil\frac{i}{2}\right\rceil,z_1^{\frac{N}{2}},v_1^{\left\lceil\frac{i}{2}\right\rceil}\right)$$

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

- 似然比递归计算详解：
  - ✓ 递归公式：

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

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

其中

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

$$f\left(N,i,y_1^N,u_1^{i-1}\right)\rightarrow f\left(\frac{N}{2},\left\lceil\frac{i}{2}\right\rceil,z_1^{\frac{N}{2}},v_1^{\left\lceil\frac{i}{2}\right\rceil}\right)$$

  - ✓返回条件：

$$f\left(1,1,y_i,\phi\right)=L_1^{(1)}\left(y_i\right)=\ln\frac{\Pr\left\{x_i=0\middle|y_i\right\}}{\Pr\left\{x_i=1\middle|y_i\right\}}=\frac{4y_i}{N_0}$$

$$\because y=1-2x+n\text{且}p(n)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(n-m)^2}{2\sigma^2}\right)$$

$$\therefore\begin{cases}p\left(y\middle|x=0\right)=\dfrac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\dfrac{(y-1)^2}{2\sigma^2}\right)\\ p\left(y\middle|x=1\right)=\dfrac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\dfrac{(y+1)^2}{2\sigma^2}\right)\end{cases}$$

$$\because p\left(x\middle|y\right)=\frac{p\left(y\middle|x\right)p(x)}{p(y)}$$

$$\therefore\ln\frac{\Pr\left\{x=0\middle|y\right\}}{\Pr\left\{x=1\middle|y\right\}}=\ln\frac{p\left(y\middle|x=0\right)p(x=0)}{p\left(y\middle|x=1\right)p(x=1)}$$

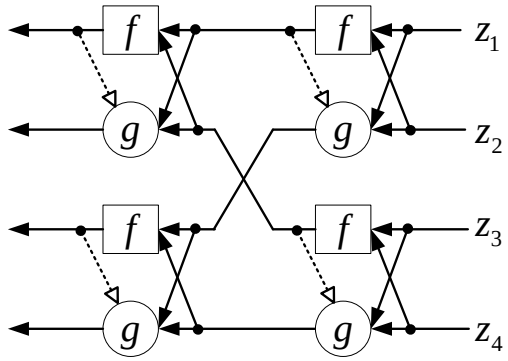
$$=\ln\frac{p\left(y\middle|x=0\right)}{p\left(y\middle|x=1\right)}=\frac{(y+1)^2-(y-1)^2}{2\sigma^2}$$

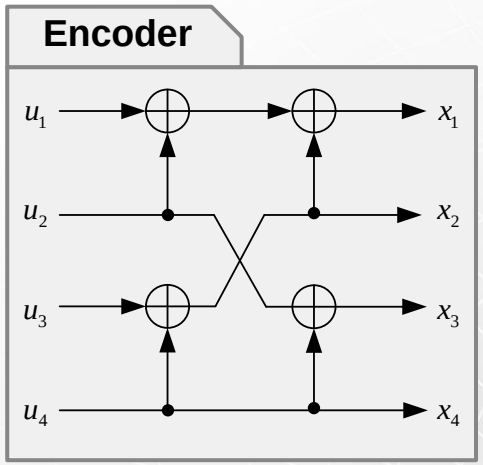
$$=\frac{2y}{\sigma^2}=\frac{4y}{N_0}$$

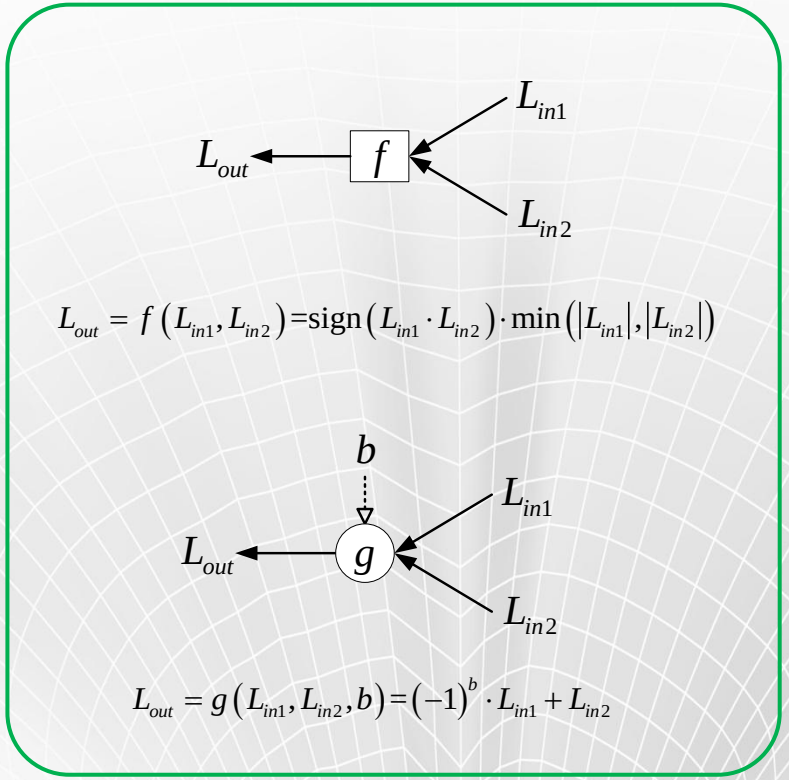
Decoding Algorithm for Polar Codes

How to calculate the path metrics?

➢在译码时可以用下图结构进行路径度量值的计算：





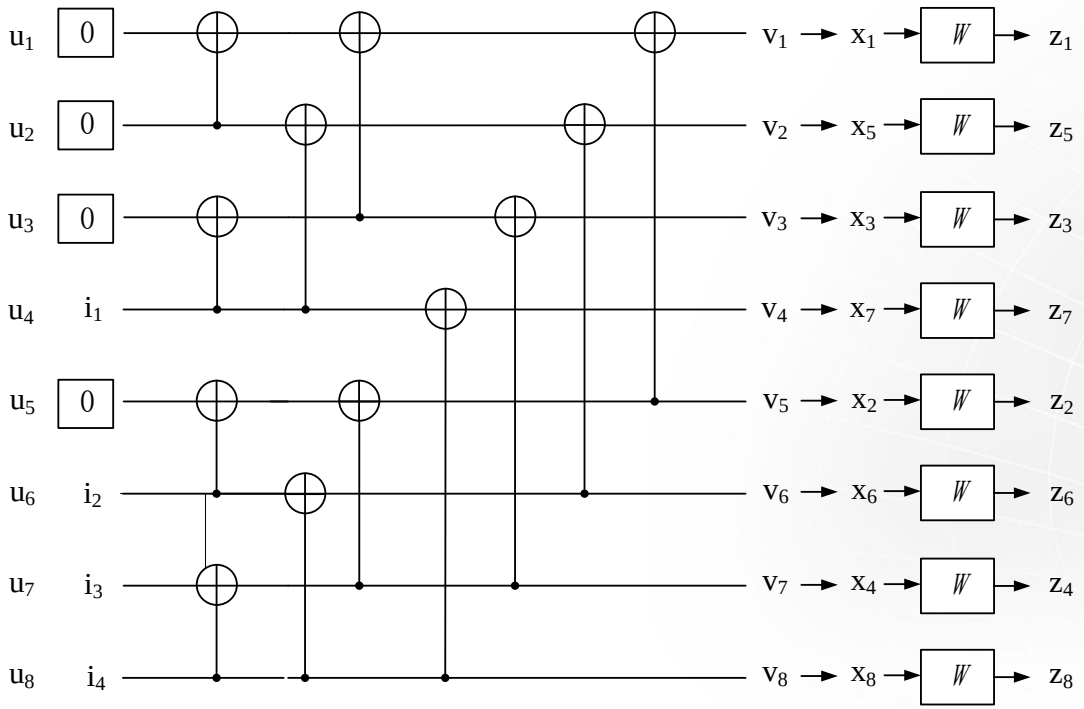


- 将编码结构中的模二加 ⊕ 和链接点 · 分别用 f 和 g 替换；
- 反向信号流方向：编码时从u到x，译码时候从z（对应x）到u。

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

➢在译码时可以用下图结构进行路径度量值的计算：



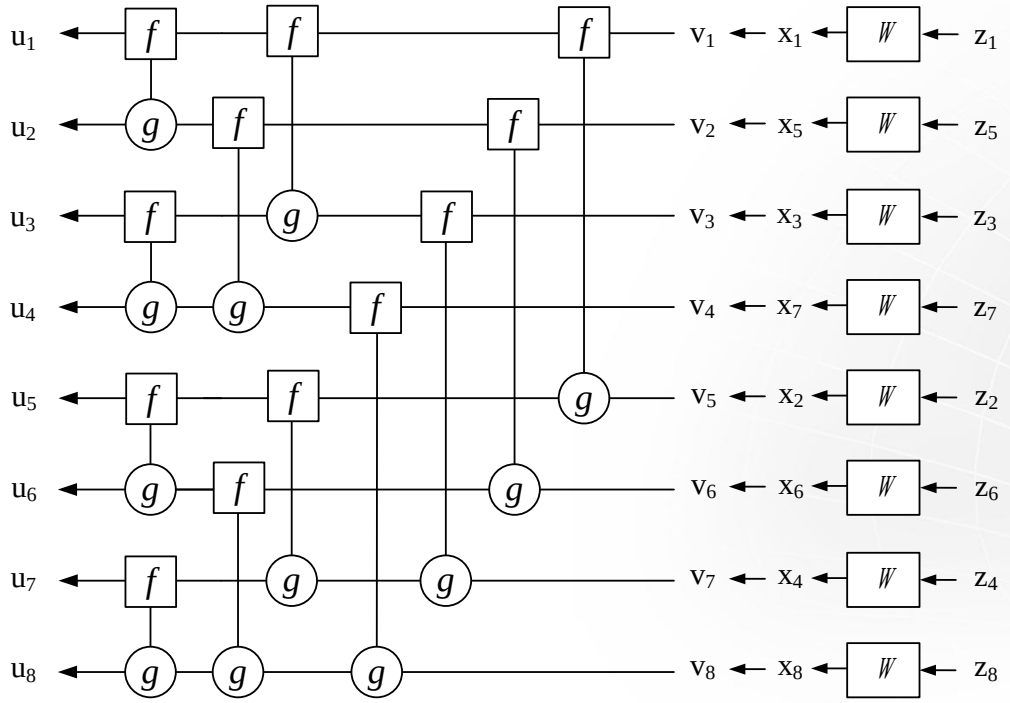
■ 将编码结构中的模二加 ⊕ 和连接点・分别用 f 和 g 替换；

■ 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。

Decoding Algorithm for Polar Codes

How to calculate the path metrics?

➢在译码时可以用下图结构进行路径度量值的计算：



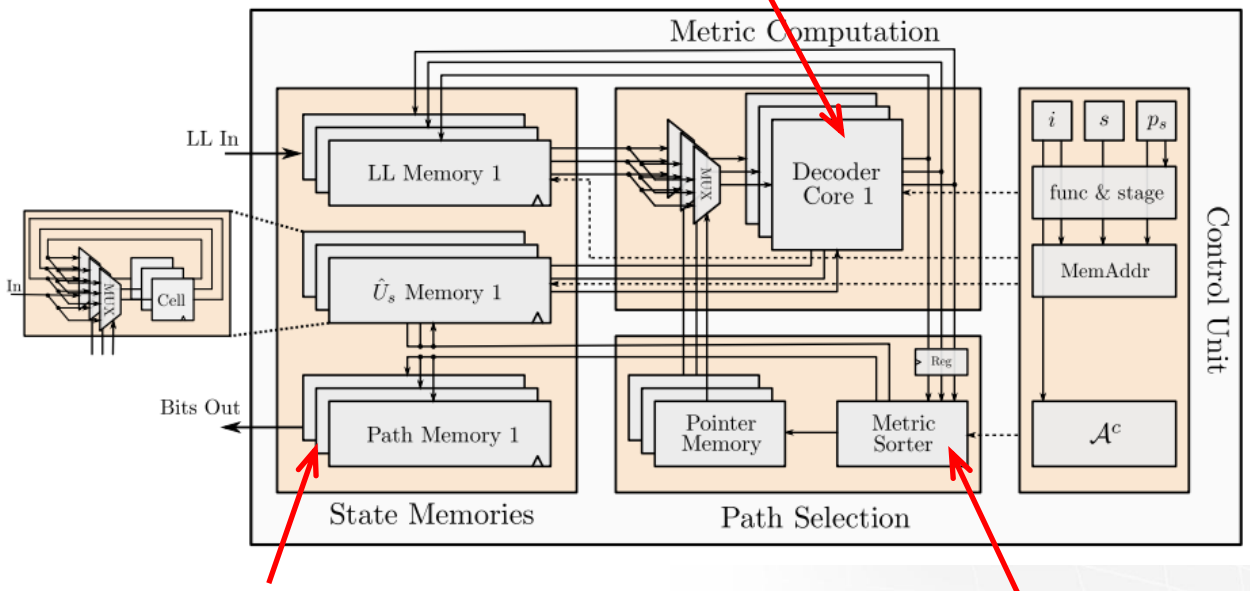
■ 将编码结构中的模二加 ⊕ 和连接点・分别用 f 和 g 替换；

■ 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。

Decoding Algorithm for Polar Codes

Summary

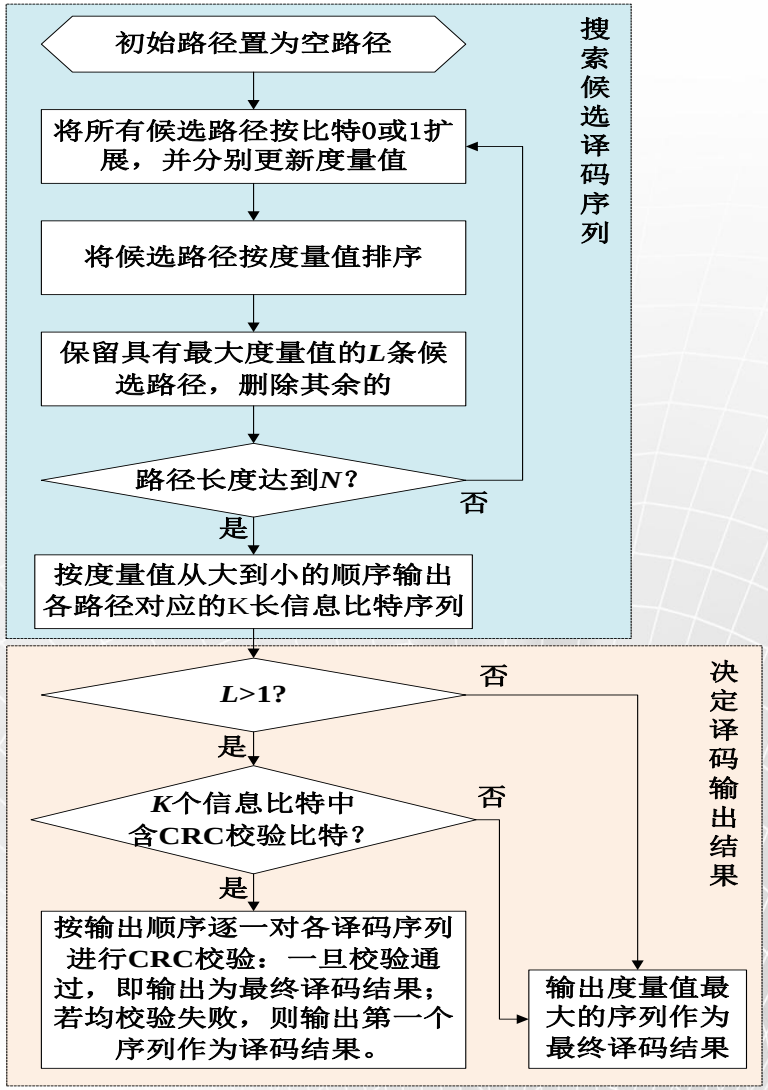
F/G computation, and path extension



Memory management for surviving Paths

Path sorting and pruning

[Fig: A. Burg, 2014]



Performance and latency

A SOFTWARE DEMO

Software Demo

Windows 版本

Windows 7 专业版

版权所有 © 2009 Microsoft Corporation。保留所有权利。

Service Pack 1

获取新版本的 Windows 7 的更多功能

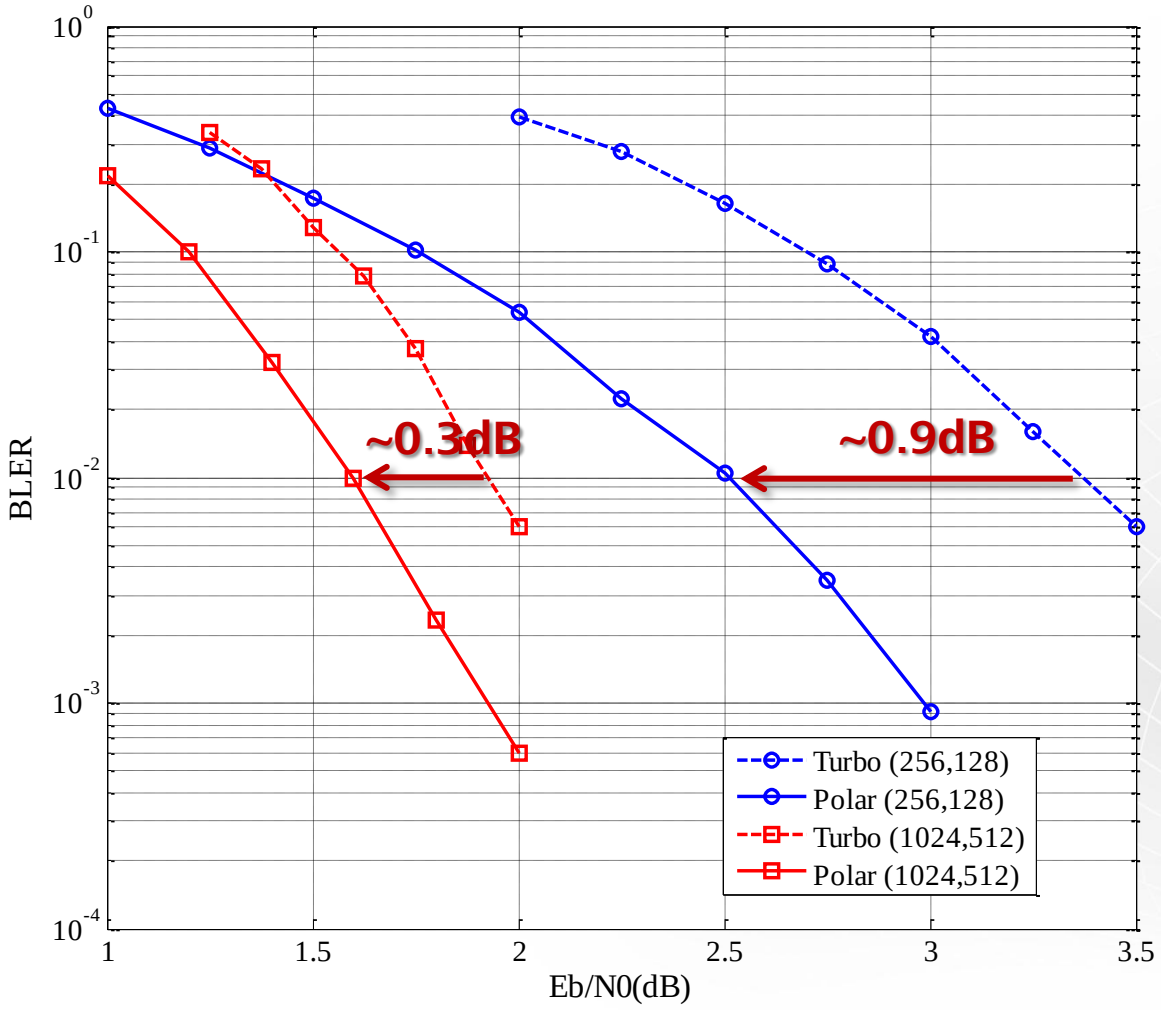
系统

分级:	系统分级不可用
处理器:	Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GHz 3.00 GHz
安装内存(RAM):	3.99 GB (3.50 GB 可用)
系统类型:	32 位操作系统
笔和触摸:	没有可用于此显示器的笔或触控输入

计算机名称、域和工作组设置

```
------------------------------------
> SNR = 1.50 dB
> N = 1024, K = 512, L=32
> Enable CRC-aided Dec = 1
Tot Blk = 4500, ErrBlk = 87
BLER = 1.93e-002, BER = 4.52e-003
Tot Time = 56.92 sec (12.65ms/blk)
------------------------------------
------------------------------------
> SNR = 1.50 dB
> N = 1024, K = 512, L=32
> Enable CRC-aided Dec = 1
Tot Blk = 4600, ErrBlk = 87
BLER = 1.89e-002, BER = 4.42e-003
Tot Time = 58.14 sec (12.64ms/blk)
------------------------------------
------------------------------------
> SNR = 1.50 dB
> N = 1024, K = 512, L=32
> Enable CRC-aided Dec = 1
Tot Blk = 4788, ErrBlk = 87
BLER = 1.85e-002, BER = 4.33e-003
Tot Time = 59.36 sec (12.63ms/blk)
------------------------------------
```

Software Demo



- BPSK+AWGN性能；
- 逐信噪比构造；


ALTERA.
UNIVERSITY
PROGRAM
THANK YOU
主办单位 |
ALTERA. UNIVERSITY PROGRAM
西安电子科技大学 XIDIAN UNIVERSITY
terasIC Terasic Inc.
赞助厂商 |
HUAWEI
intel
SPREADTRUM