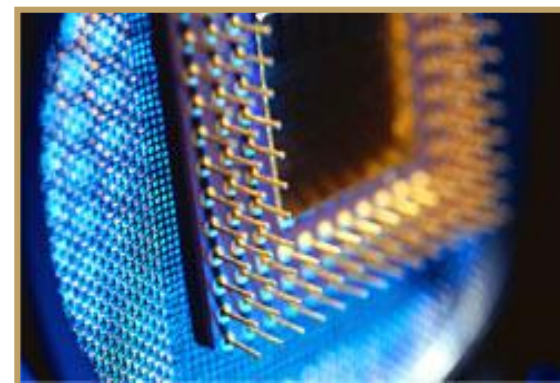
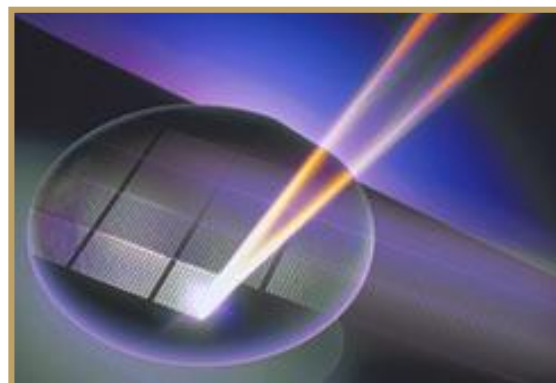
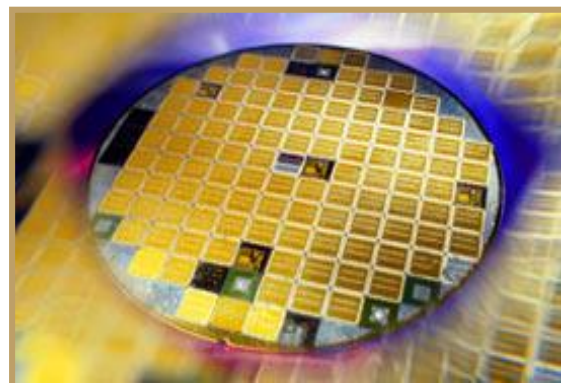




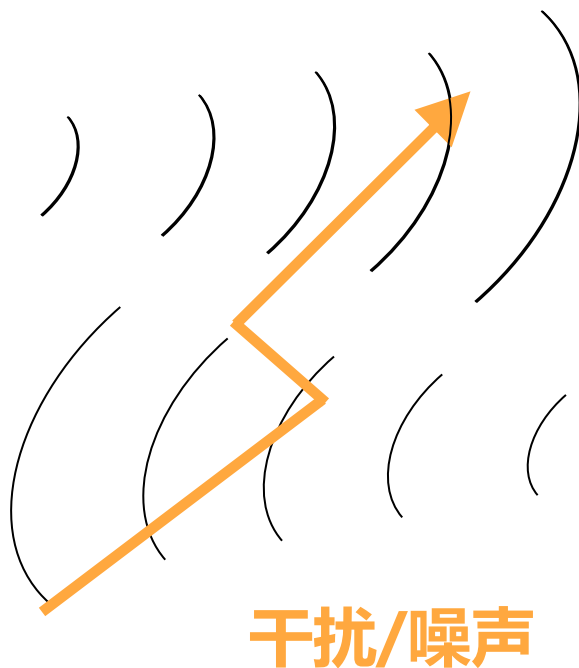
《VLSI数字通信原理与设计》课程

第十三讲：卷积编码与译码



语音通信中的干扰与噪声

010101010101



011101110101

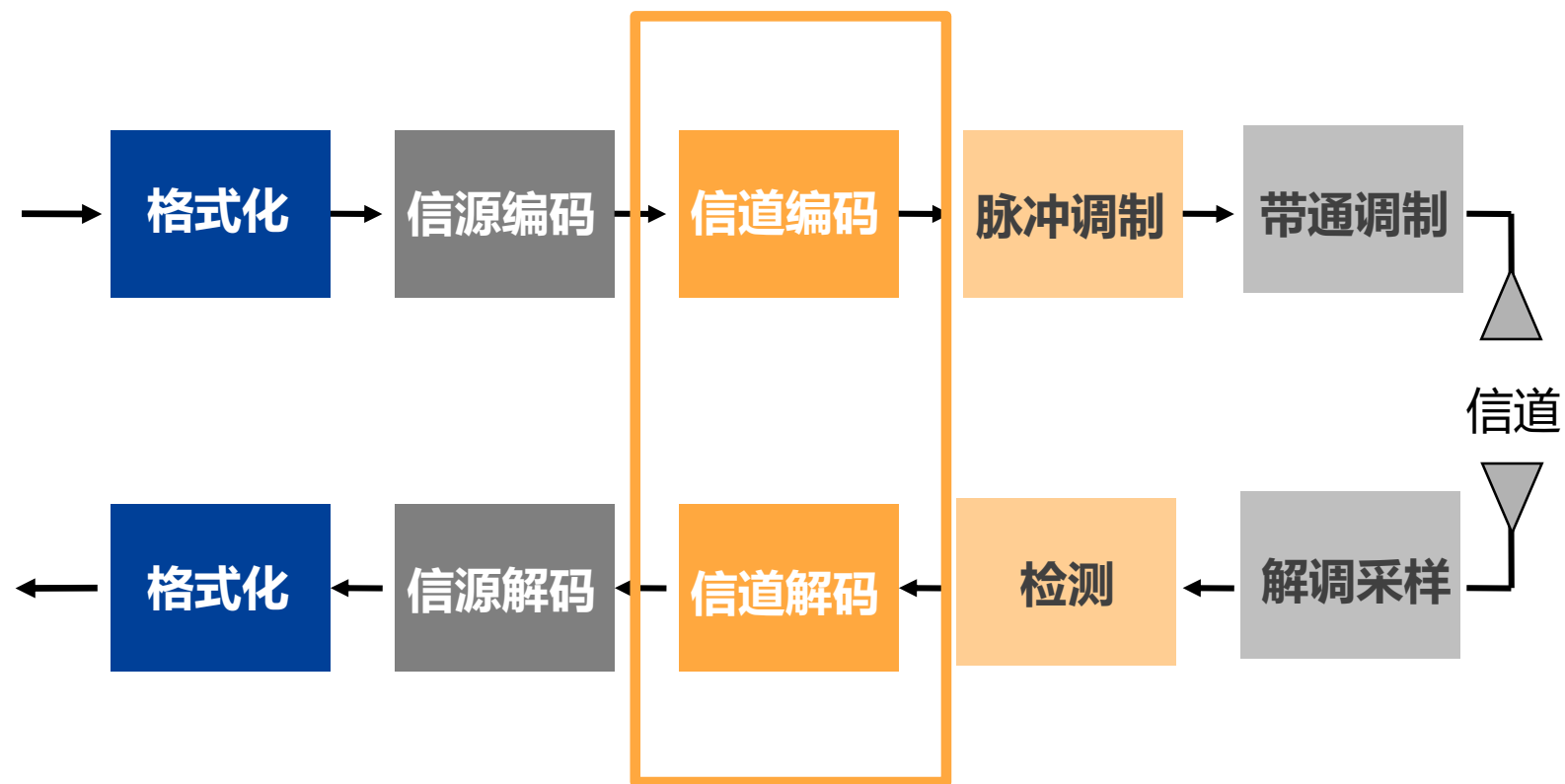


如何纠正传输错误，提高通信系统可靠性



信道编码

信道编码在通信系统中的位置



目的：提高信号传输的可靠性

方法：增加冗余比特，以发现或纠正错误

编码方式：卷积码

卷积码的应用





目录

01 卷积码的基本概念

02 卷积码的编码方法

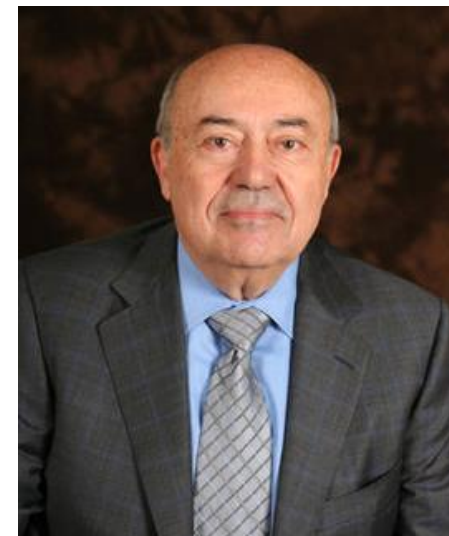
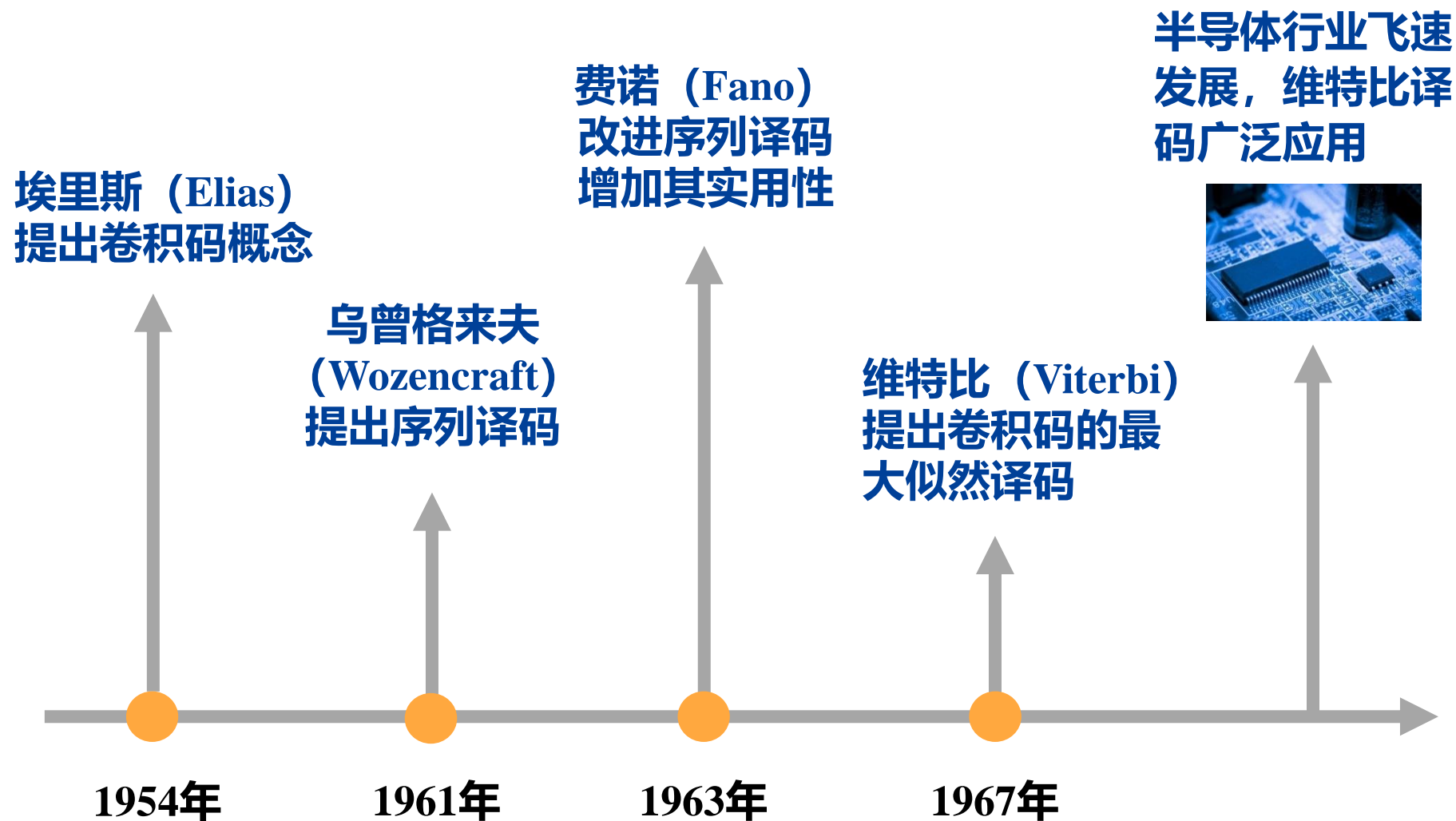
• **03** 卷积码的译码方法

04 卷积码的性能

05 总结



01. 卷积码的基本概念 —— 发展历史



维特比：
高通公司创始人之一，
维特比算法提出者



目 录

01 卷积码的基本概念

02 卷积码的编码方法

• 03 卷积码的译码方法

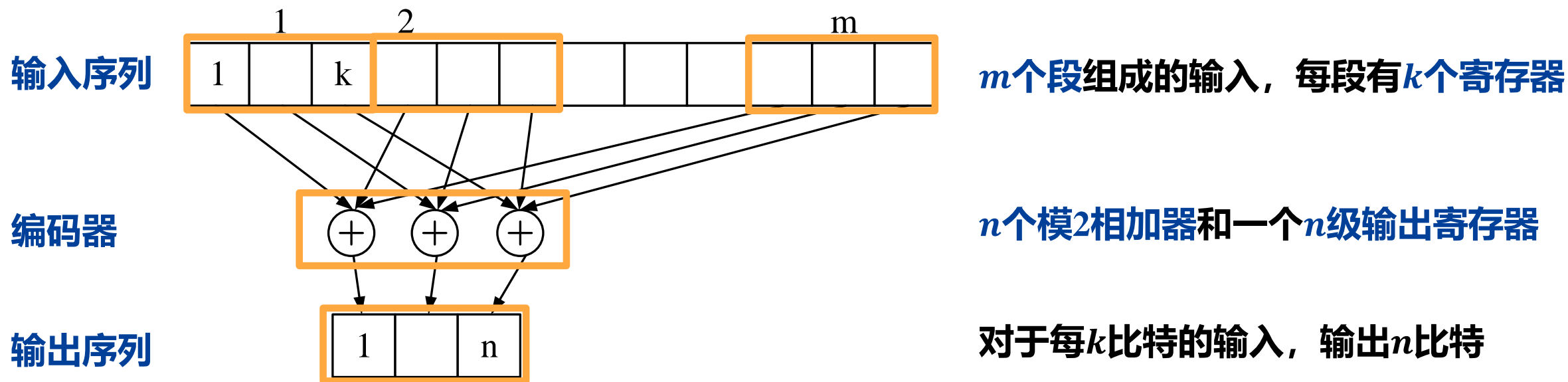
04 卷积码的性能

05 总结



02. 卷积码的编码方法 —— 基本概念

卷积编码：通过将输入信息序列与编码器做卷积运算，将 k 位信息编成 n 比特。
此 n 比特不仅与当前 k 位信息有关，还与前面 $(m - 1)$ 段的信息有关。

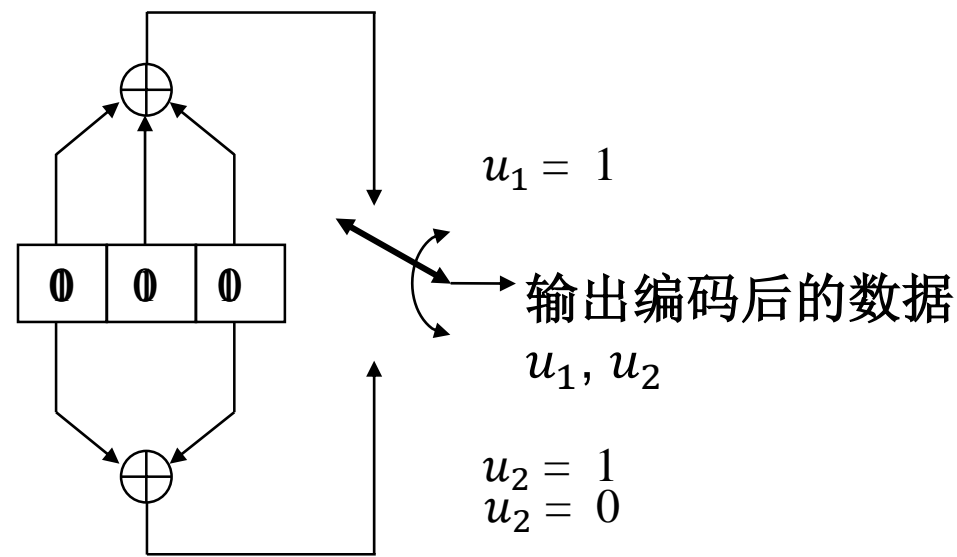


卷积码 (n, k, m) 的参数：

- k : 输入比特信息
- n : 输出码字
- m : 约束长度，即移位寄存器的级数
- 码率 R : 传输信息的有效性, $R = k/n$

02.卷积码的编码方法 —— 冲激响应

■ 输入 “1” 对应的输出序列为编码器的响应



寄存器 内容	分支编码输出	
	u_1	u_2
100	1	1
010	1	0
001	1	1

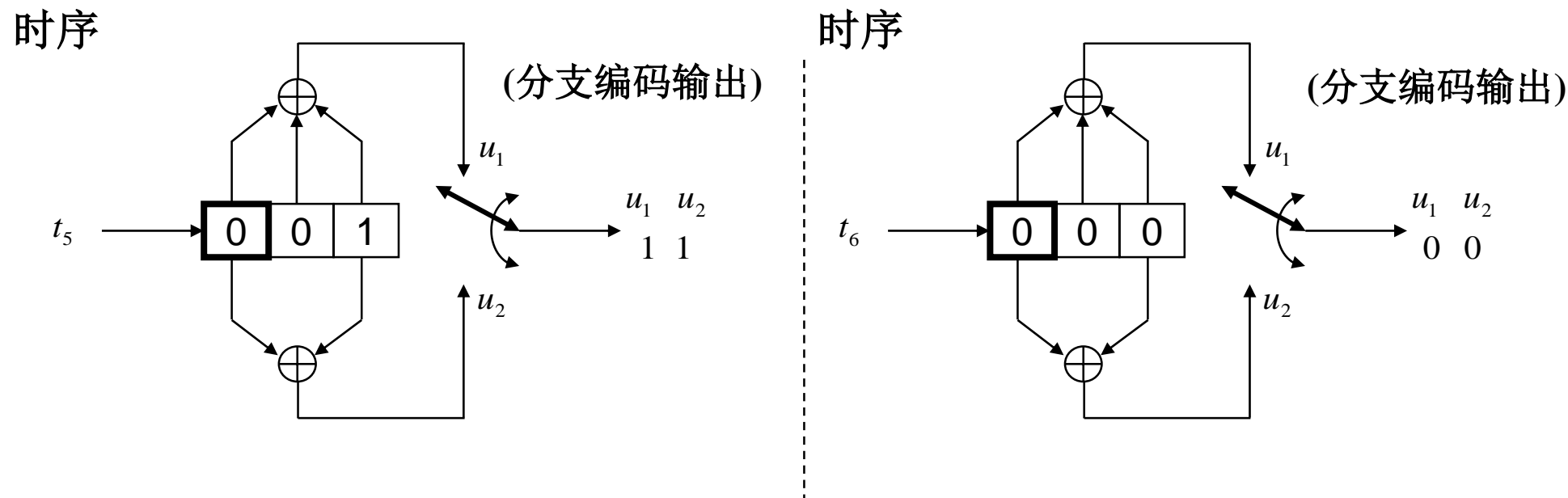
输入序列	1	0	0
输出序列	11	10	11

02. 卷积码的编码方法 —— 冲激响应

输入序列 $m = 1\ 0\ 1$ 对应的输出可按如下的线性叠加方式而得到

输入 m	输出
1	1 1 1 0 1 1
0	0 0 0 0 0 0
1	1 1 1 0 1 1
模2求和:	1 1 1 0 0 0 1 0 1

02. 卷积码的编码方法 —— 冲激响应



$m = (1\ 0\ 1) \longrightarrow$ **卷积编码** $\longrightarrow U = (11\ 10\ 00\ 10\ 11)$

3-bit 输入, 10-bit输出, 有效编码效率 $k/n = 3/10$

为什么不是1/2?

如果信息序列较长, 比如300 bits, 输出码字序列为604 bits, 编码效率300/604—更接近1/2

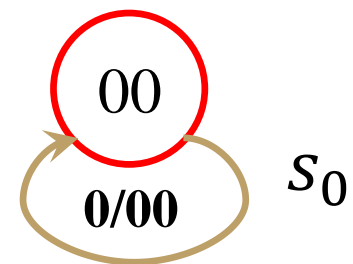
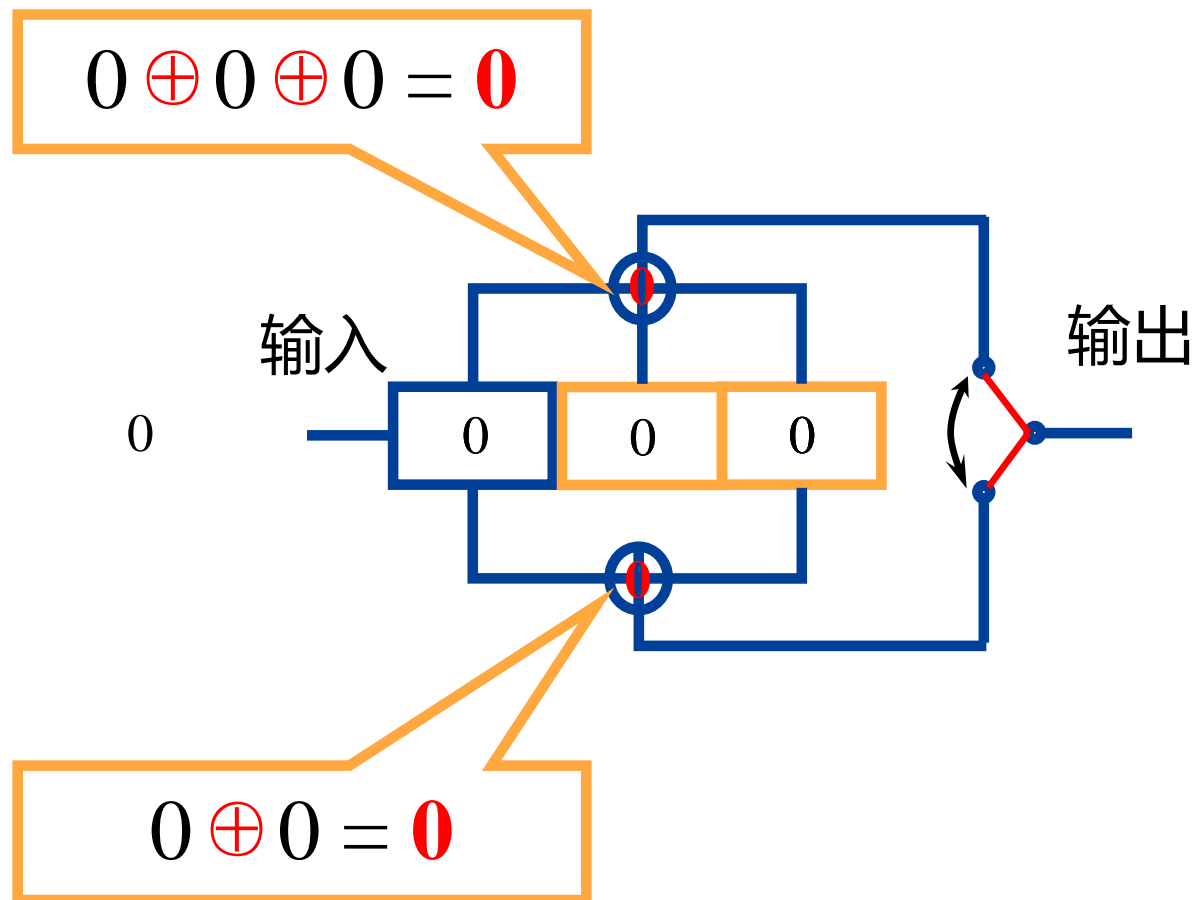
02. 卷积码的编码方法 —— 状态描述和状态图

- 有限状态机仅包含有限状态数

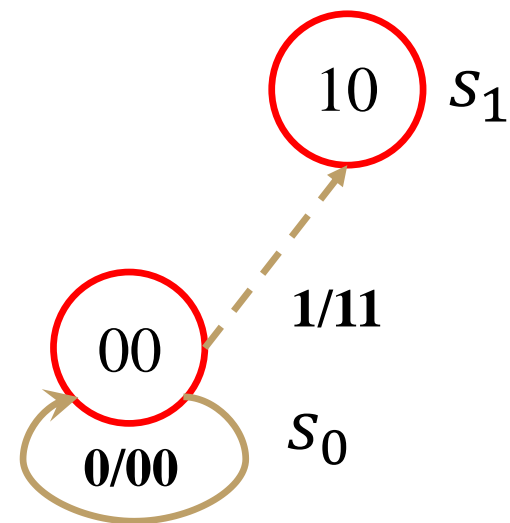
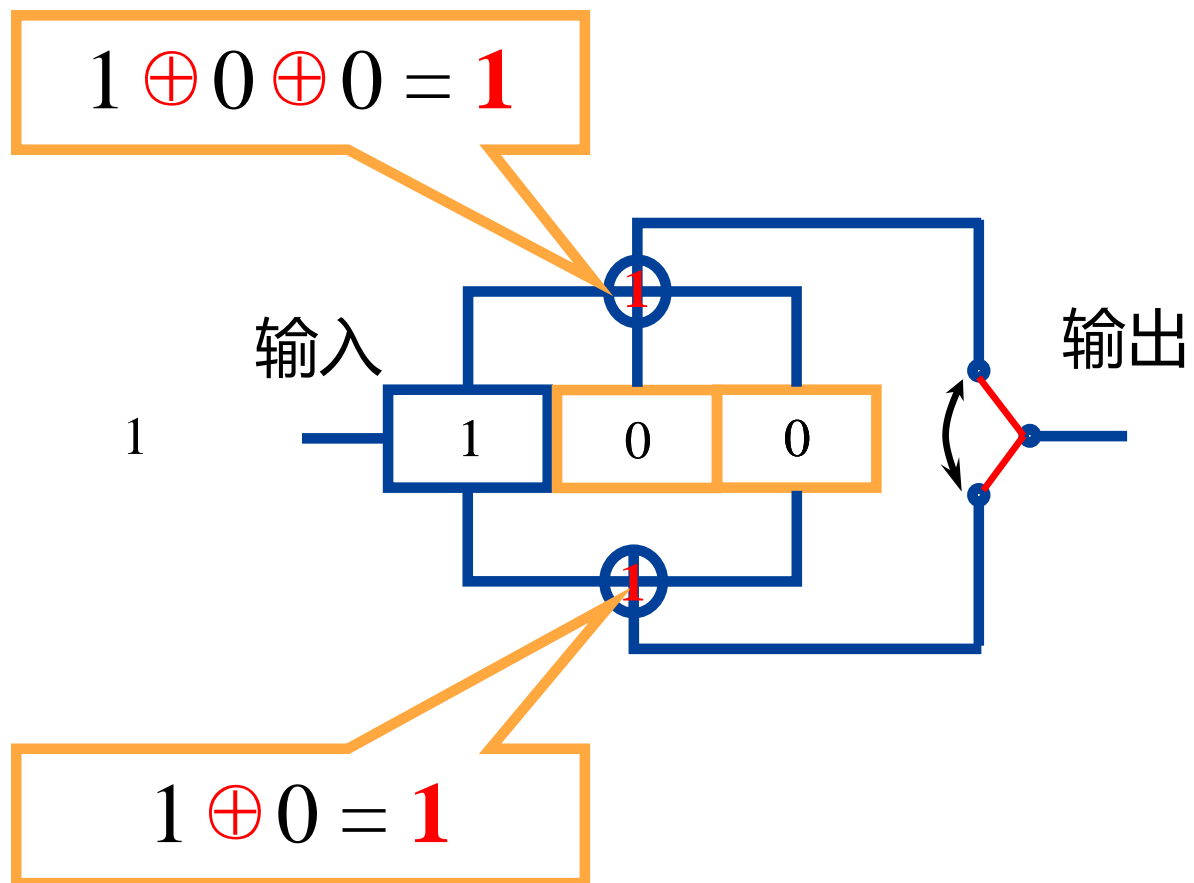
- 可以用当前输入和之前 $(K-1)$ 个输入，来预测输出

- 卷积编码器中，状态共有 2^{K-1} 个

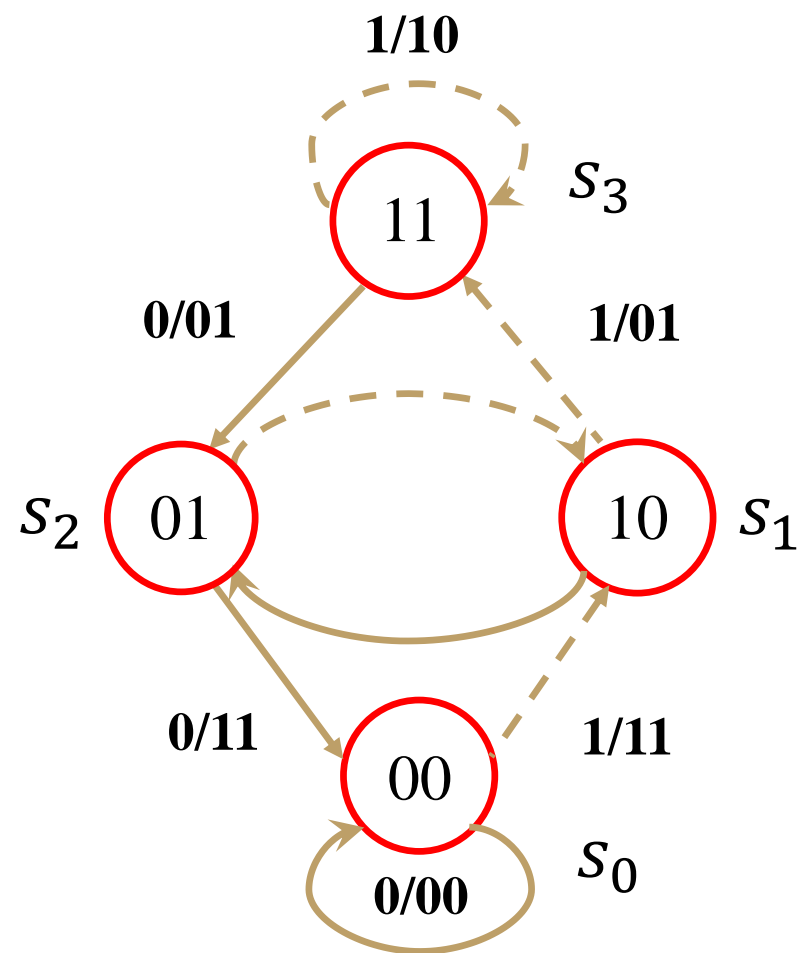
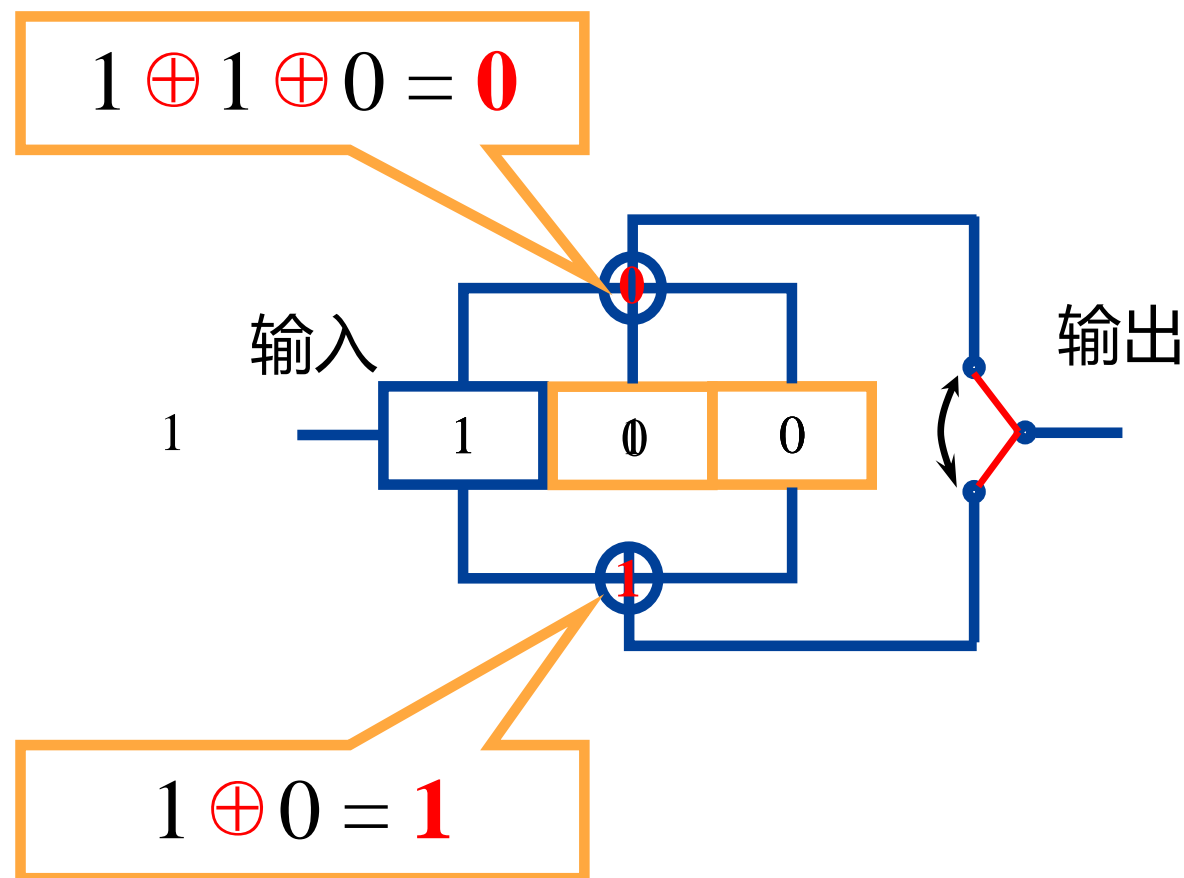
02. 卷积码的编码方法 —— (2,1,3) 卷积码的编码方法



02. 卷积码的编码方法 —— (2,1,3) 卷积码的编码方法

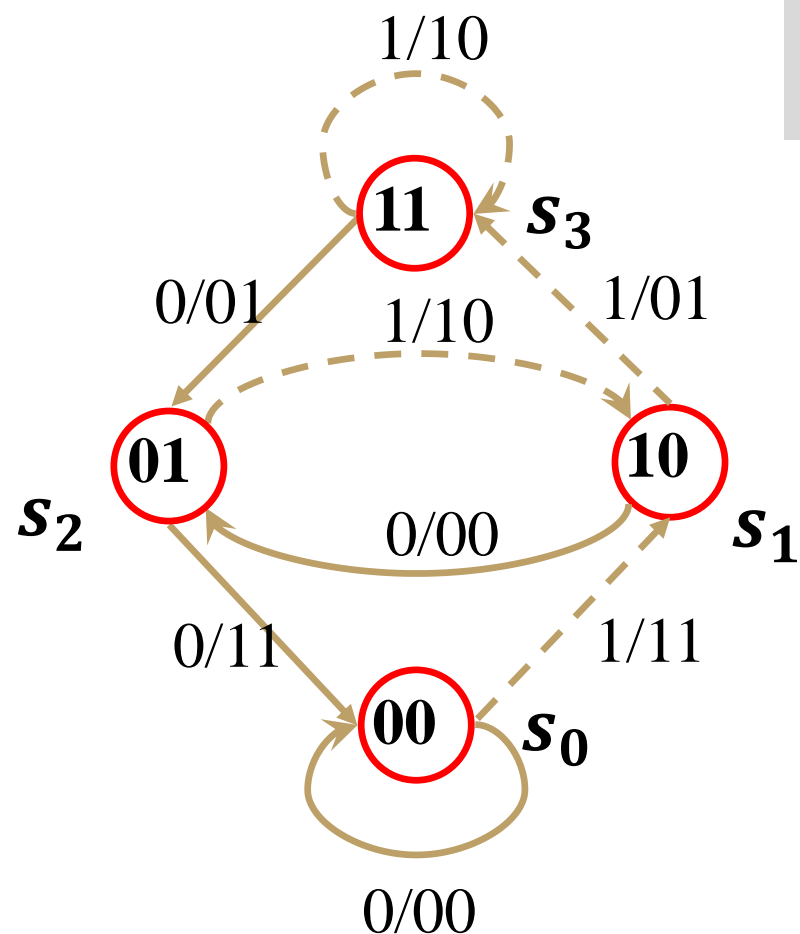


02. 卷积码的编码方法 —— (2,1,3) 卷积码的编码方法



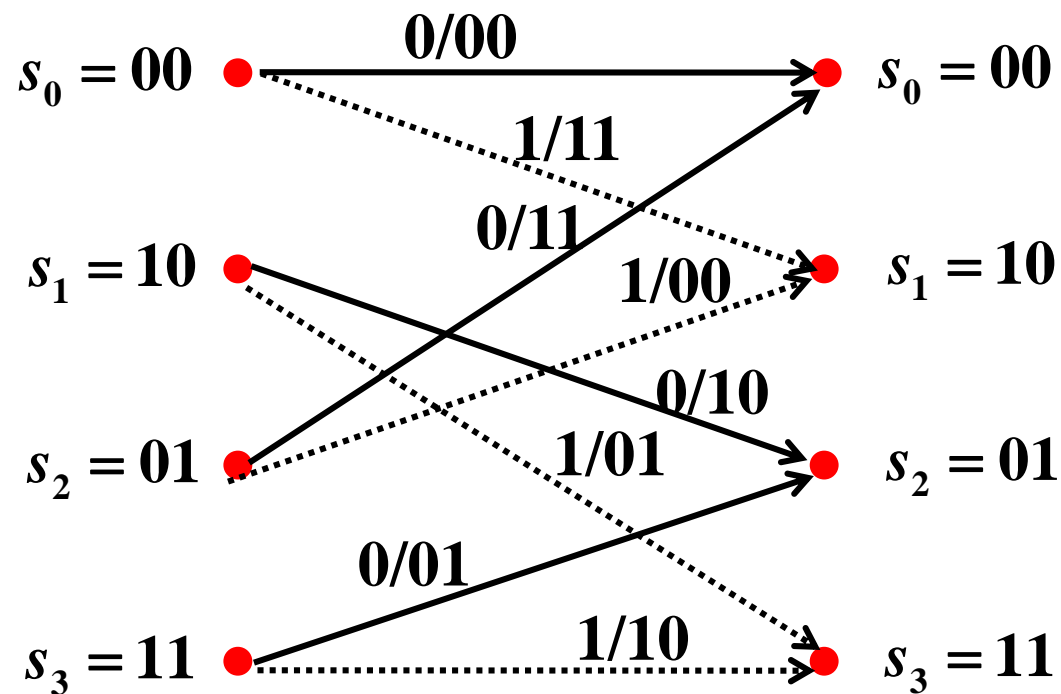
(2,1,3) 卷积编码的状态转移图

02. 卷积码的编码方法 —— (2,1,3) 卷积编码表示方法



(2,1,3) 卷积编码的状态转移图

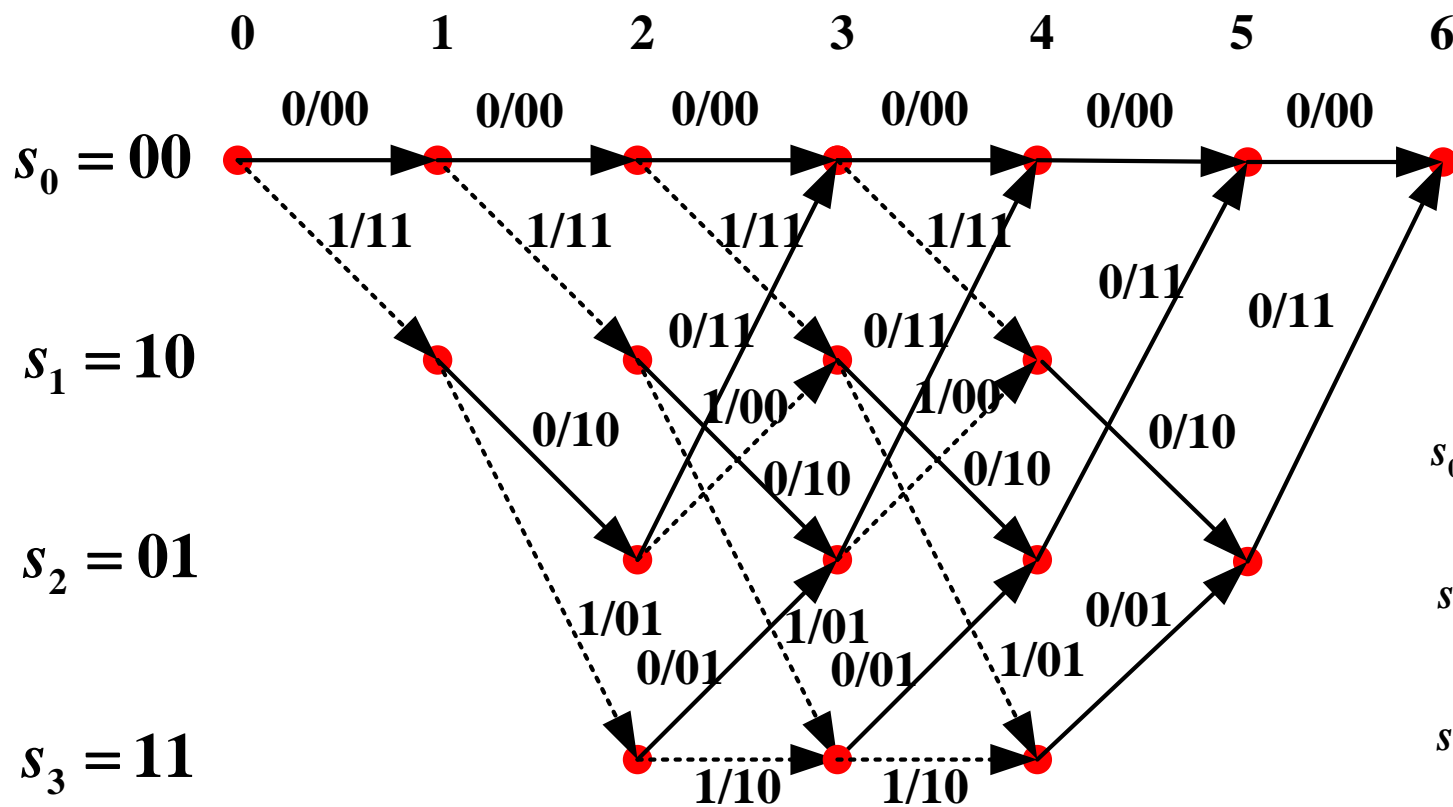
实线表示输入0
虚线表示输入1



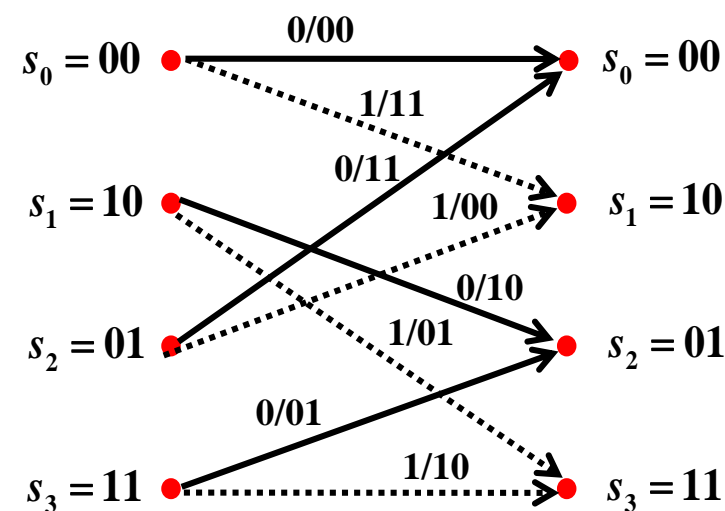
(2,1,3) 卷积编码的状态图

02. 卷积码的编码方法 —— 网格图表示

网格图：把编码器的状态图沿时间轴展开



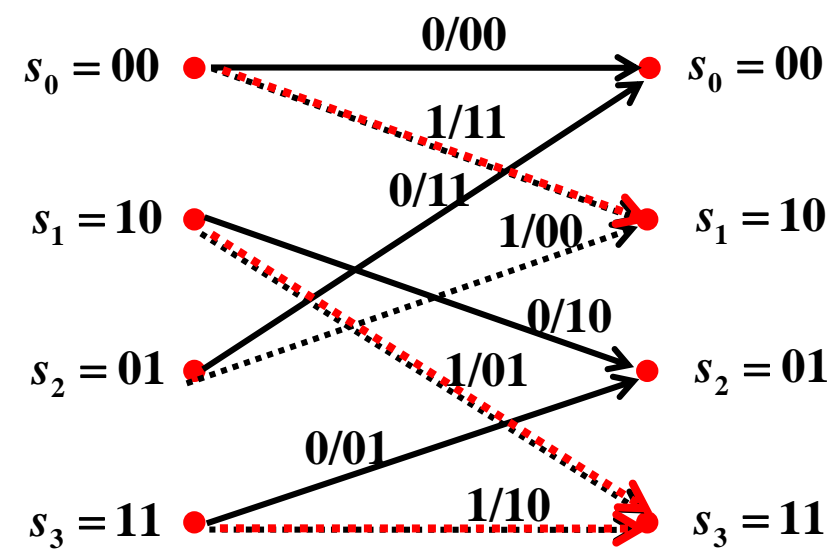
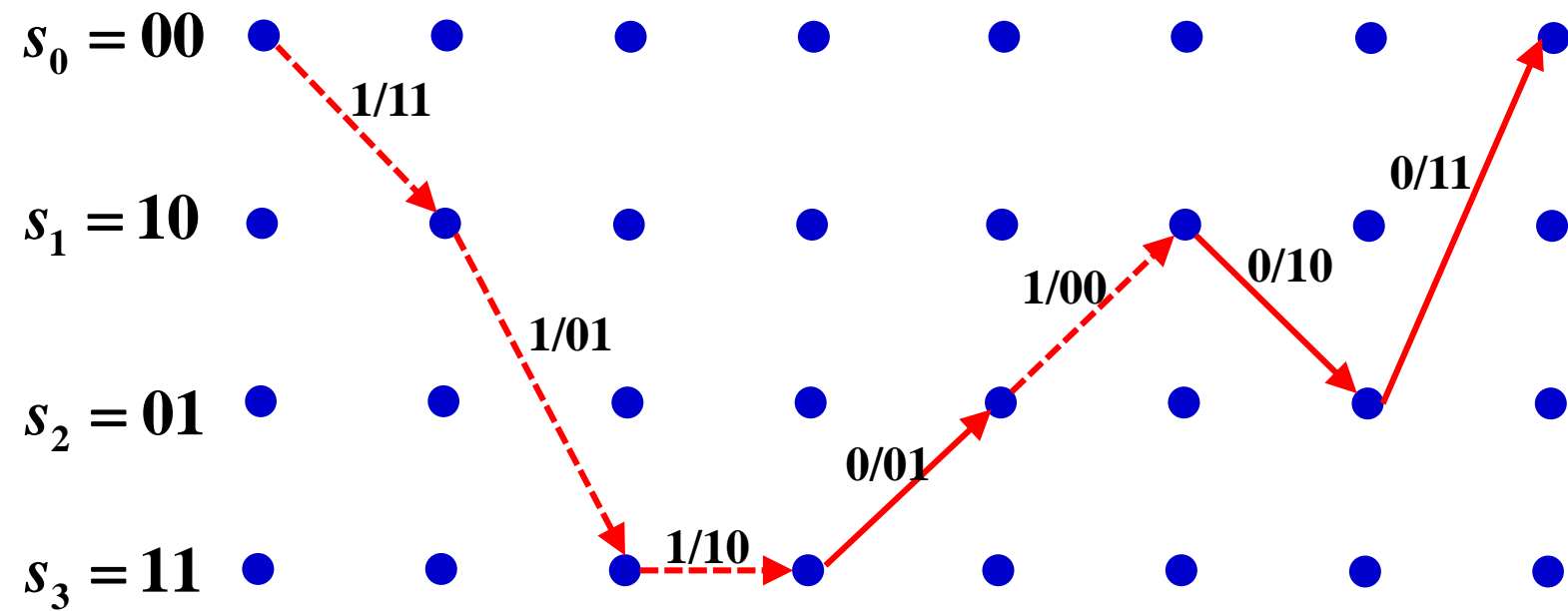
(2,1,3)卷积码的网格图



02. 卷积码的编码方法 —— 网格图分析

■ 网格图：把编码器的状态图沿时间轴展开

输入： 1 1 1 0 1 0 0
状态： s_0 s_1 s_3 s_3 s_2 s_1 s_2 s_0
输出： 11 01 10 01 00 10 11
接收： 11 01 10 01 01 10 10



(2,1,3)卷积码的状态图

如何进行正确的译码？



目 录

01 卷积码的基本概念

02 卷积码的编码方法

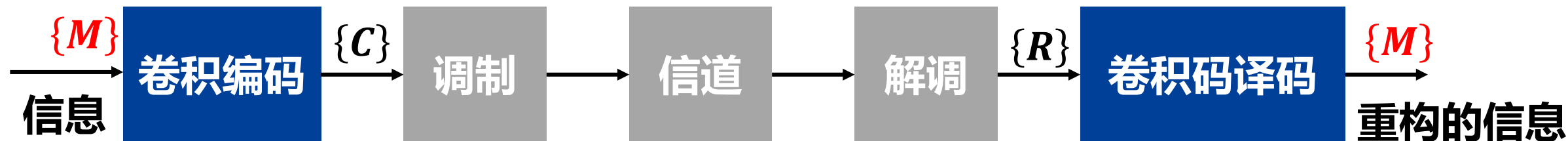
• **03** 卷积码的译码方法

04 卷积码的性能

05 总结



03. 卷积码的译码方法 —— 基本概念



译码任务

- 由接收序列 R 给出与发端信息序列 M 最接近的估值序列 \hat{M}
- 译码器根据接收 R 产生发送 C 的估值序列 \hat{C}

$$\text{当 } \hat{C} \neq C \text{ 时, } \hat{M} \neq M$$

$$P_E = \sum_R P(E|R)P(R) = \sum_R P(\hat{C} \neq C|R)P(R)$$

- $P(R)$: 接收 R 的概率, 与译码方法无关。

03. 卷积码的译码方法 —— 最大似然译码

最大似然译码：所有可能码字序列中选择似然函数最大的路径

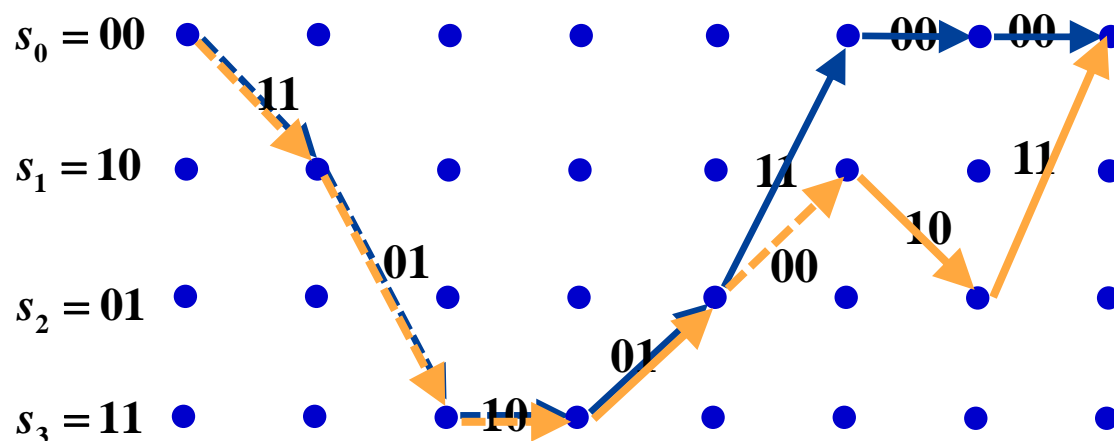
$$P(R|\hat{C}) = \max_{i=1,2,\dots,2^{kL}} P(R|C^{(i)})$$

L : 输入信息段数

$C^{(i)}$: 可能的码字序列

2^{kL} : 总路径数

最大似然译码就是寻求与接收序列具有最小汉明距离的发送序列！



- 遍历 2^{kL} 条路径，复杂度高
- 接收完整整个码字后才能译码，实时性差

如何解决？

汉明距：两个等长字符串在对应位置上不同字符的数目

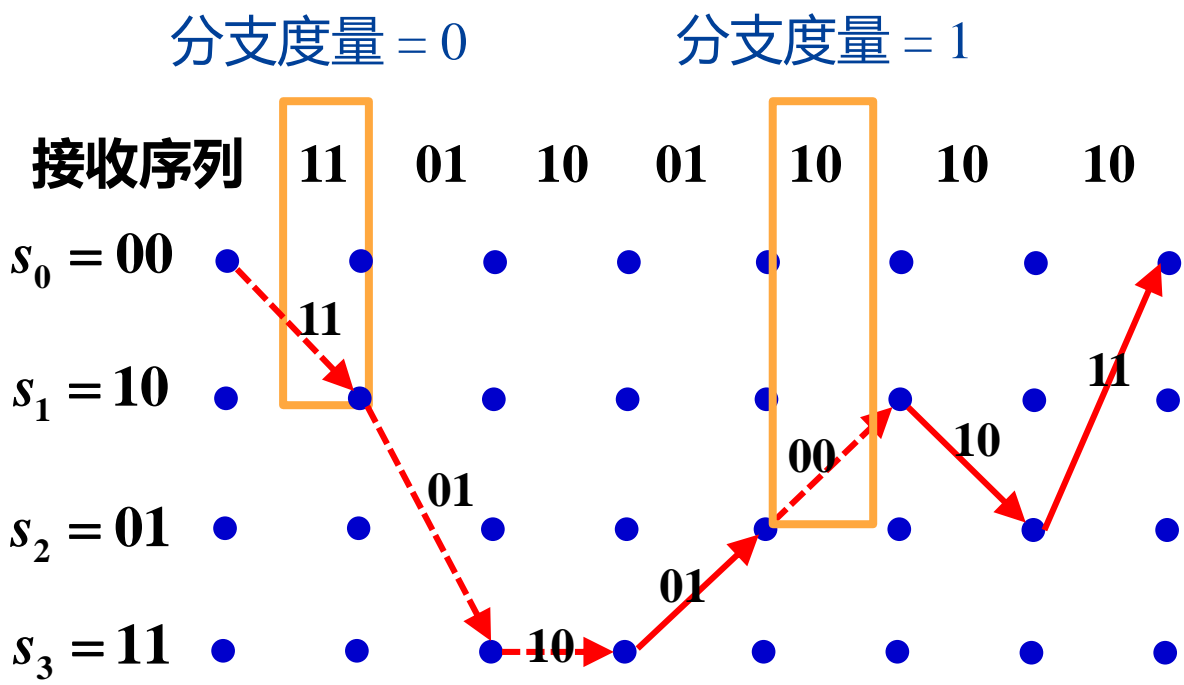
$$d = 5$$



03. 卷积码的译码方法 —— 维特比译码

维特比译码是基于动态规划的方法

- 动态规划思想：当前最优路径 = 上一步最优路径 + 当前组合
- 维特比译码思想：译码器每接收一段，就计算比较判决



路径度量 = 0 + 0 + 0 + 0 + 1 + 0 + 1 = 2

- **分支度量**：某时刻输出子码与接收子码之间的汉明距离
- **路径度量**：该路径输出序列与接收序列之间的汉明距离，是分支度量的和

03. 卷积码的译码方法 —— 维特比译码

维特比译码是基于动态规划的方法

- 动态规划思想：

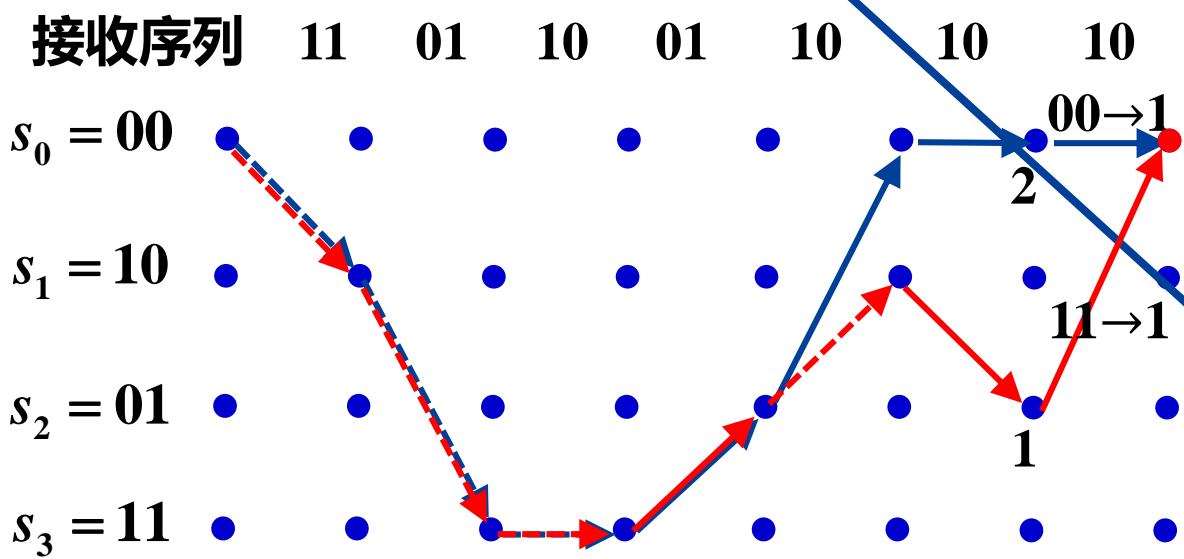
当前最优路径

 =

上一步最优路径

 +

当前组合
- 维特比译码思想：译码器每接收一段，就计算比较判决



维特比译码步骤：

- 计算到达当前节点的**分支度量**；
- 取出到达上一节点对应的**路径度量**；
- 相**加**得到两个新路径度量；**比较**，**选**出汉明距离较小者，被称作幸存路径

03. 卷积码的译码方法 —— 维特比译码举例 (1)

信息序列M: 1

1

1

0

1

0

0

发送码字C: 11

01

10

01

00

10

11

接收序列R: 11

01

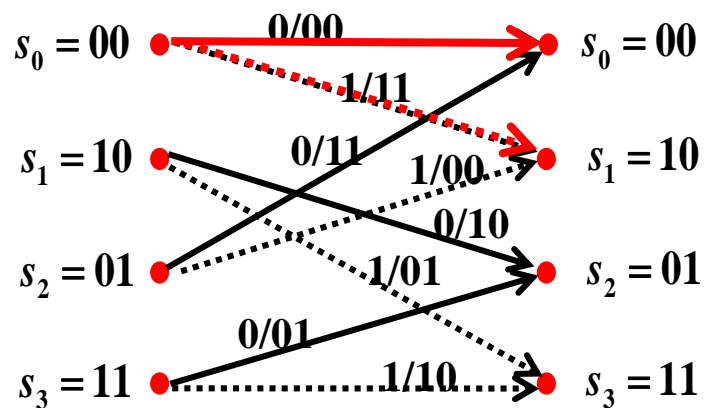
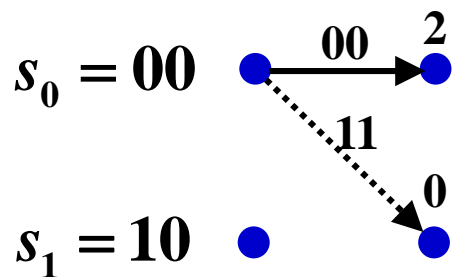
10

01

01

10

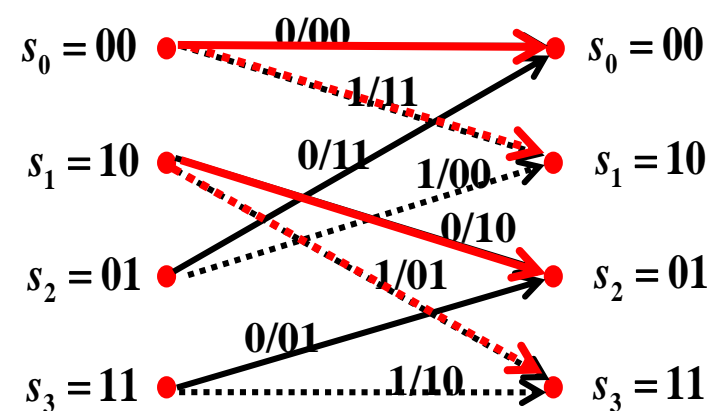
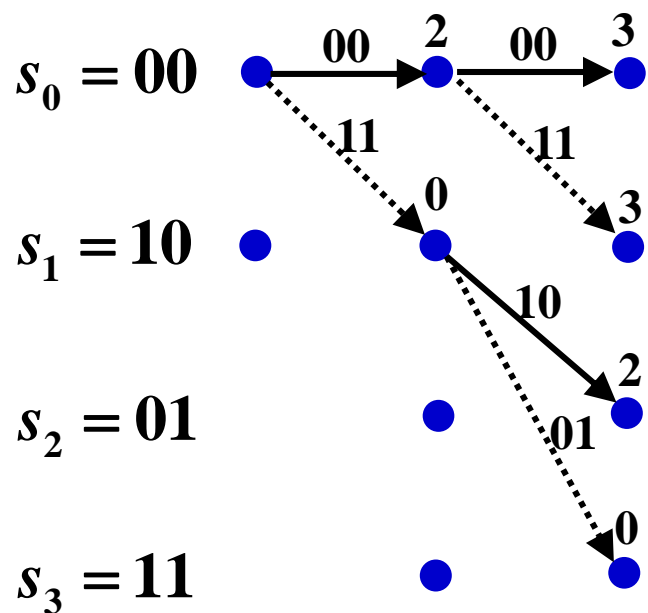
10



补零

03. 卷积码的译码方法 —— 维特比译码举例 (2)

信息序列M:	1	1	1	0	1	0	0
发送码字C:	11	01	10	01	00	10	11
接收序列R:	11	01	10	01	01	10	10

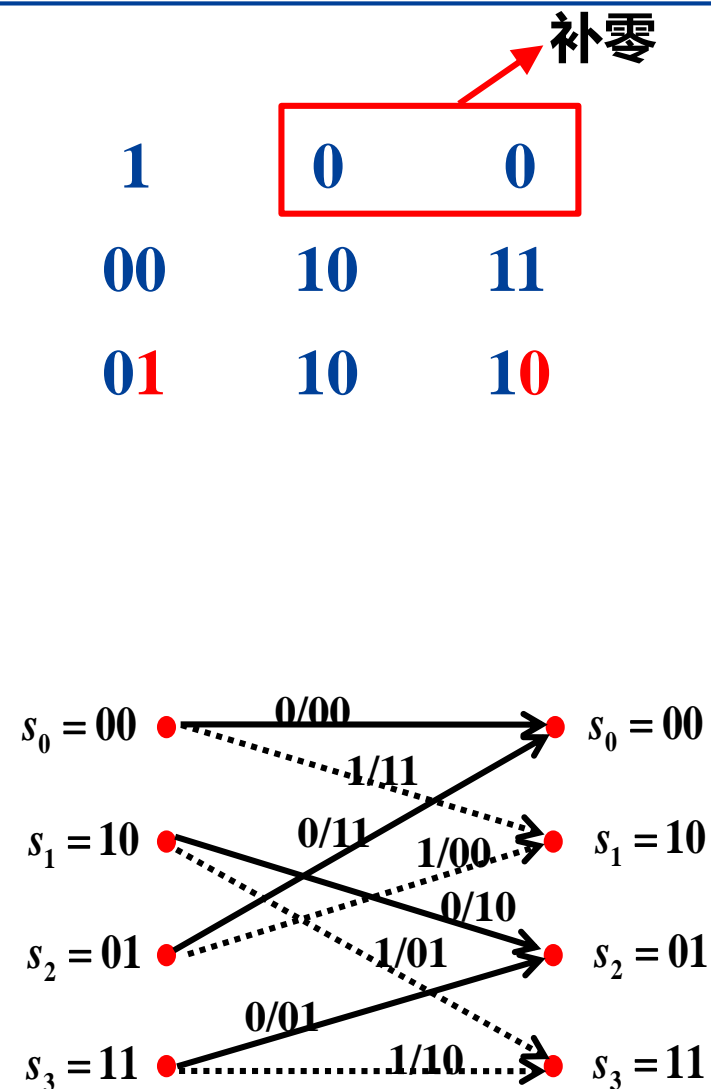
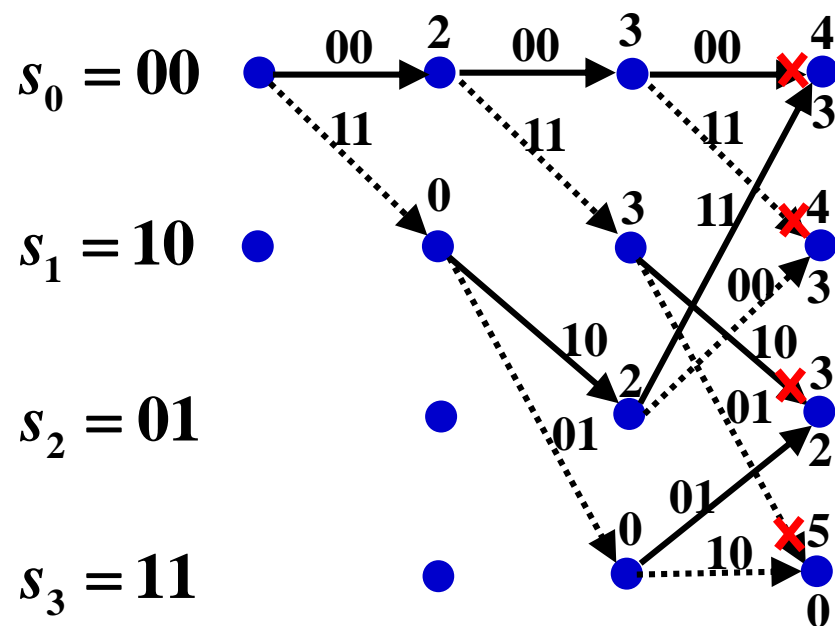


03. 卷积码的译码方法 —— 维特比译码举例 (3)

信息序列M: 1 1 1 0 1 0 0

发送码字C: 11 01 10 01 00 10 11

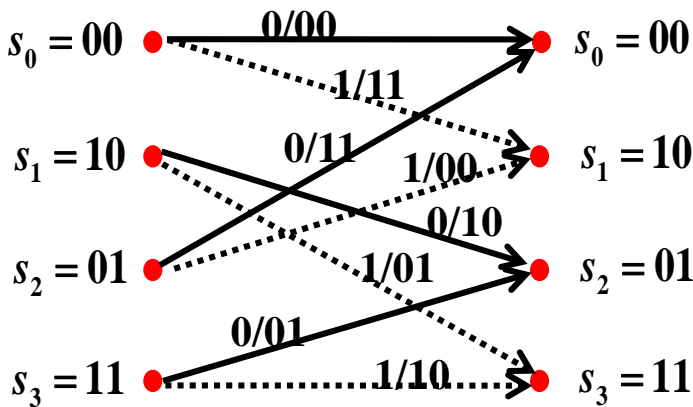
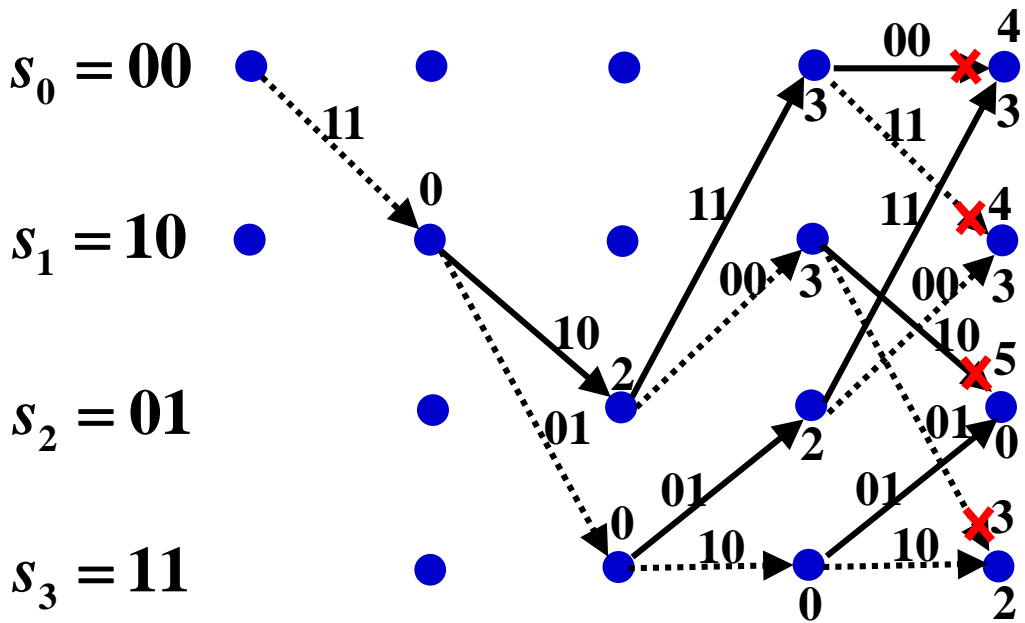
接收序列R: 11 01 10 01 01 10 10



03. 卷积码的译码方法 —— 维特比译码举例（4）

信息序列M:	1	1	1	0	1	0	0
发送码字C:	11	01	10	01	00	10	11
接收序列R:	11	01	10	01	01	10	10

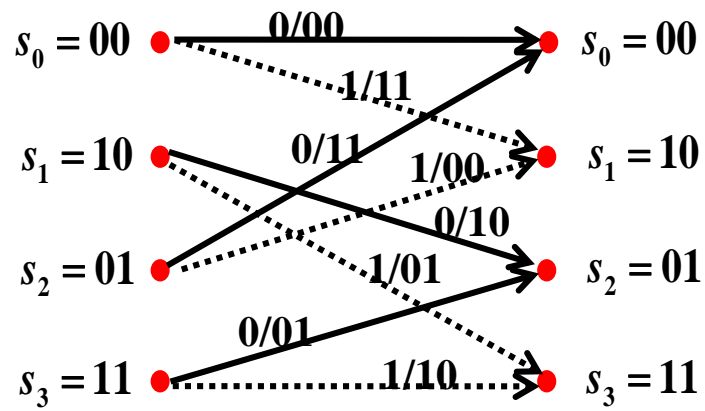
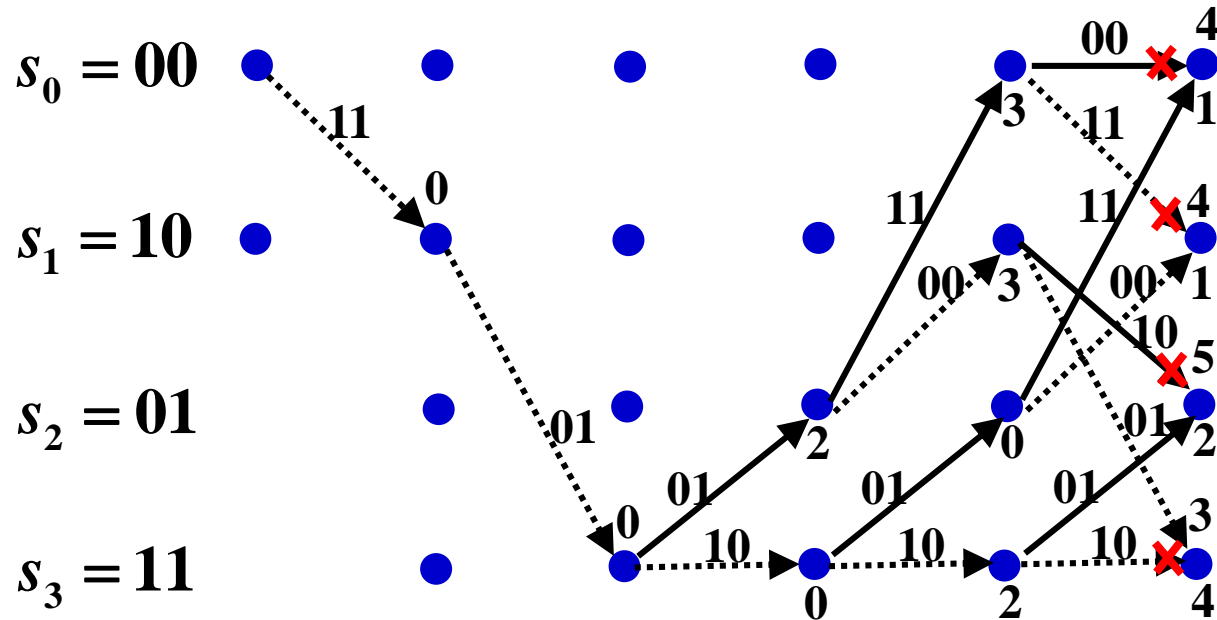
补零



03. 卷积码的译码方法 —— 维特比译码举例（5）

信息序列M:	1	1	1	0	1	0	0
发送码字C:	11	01	10	01	00	10	11
接收序列R:	11	01	10	01	01	10	10

补零



运算量和存储量与码长呈线性关系

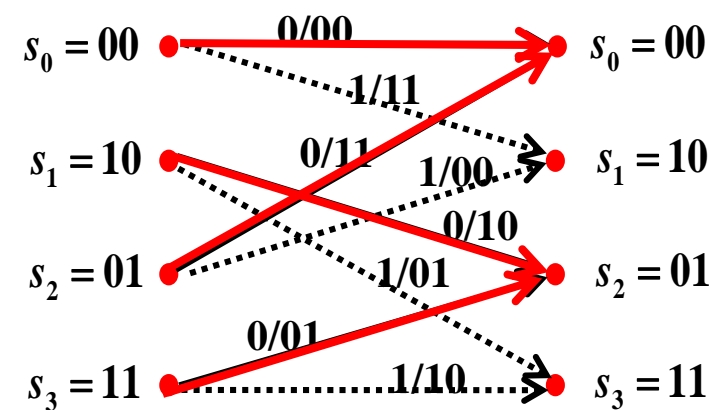
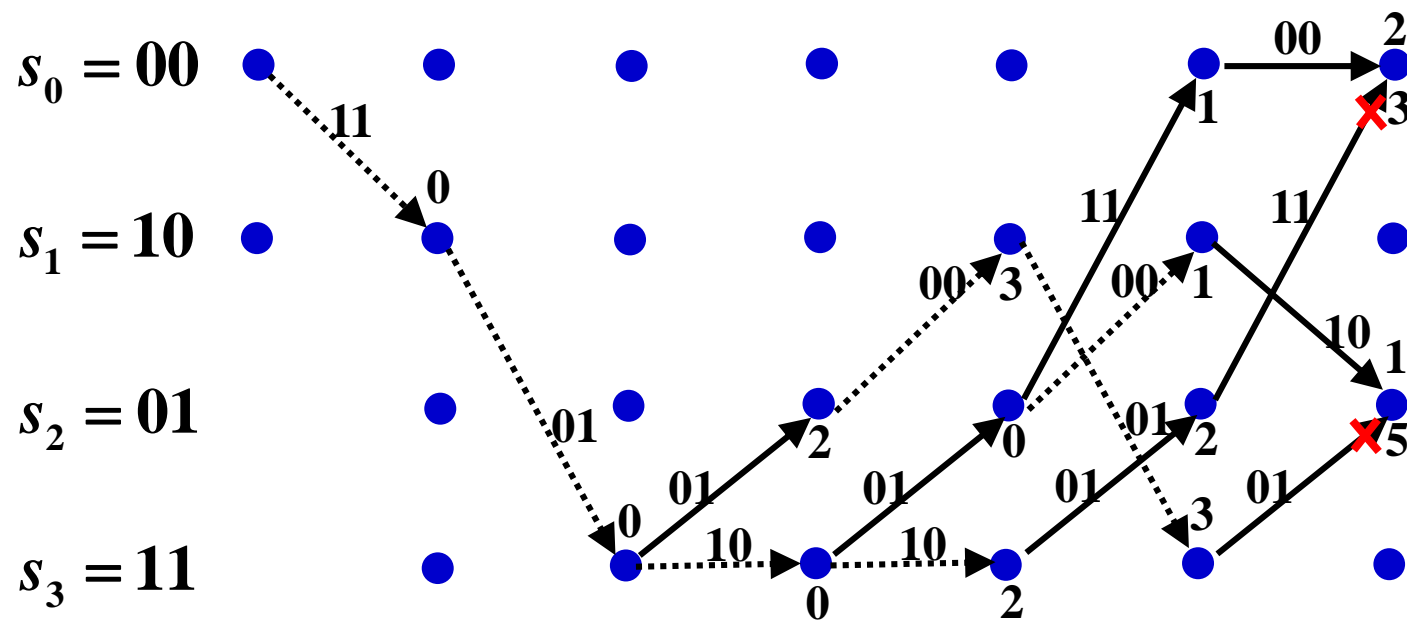
03. 卷积码的译码方法 —— 维特比译码举例（6）

信息序列M: 1 1 1 0 1

发送码字C: 11 01 10 01 00

接收序列R: 11 01 10 01 01

补零
0 0
10 11
10 10



03. 卷积码的译码方法 —— 维特比译码举例 (7)

信息序列M: 1

1

1

0

1

0

0

发送码字C: 11

01

10

01

00

10

11

接收序列R: 11

01

10

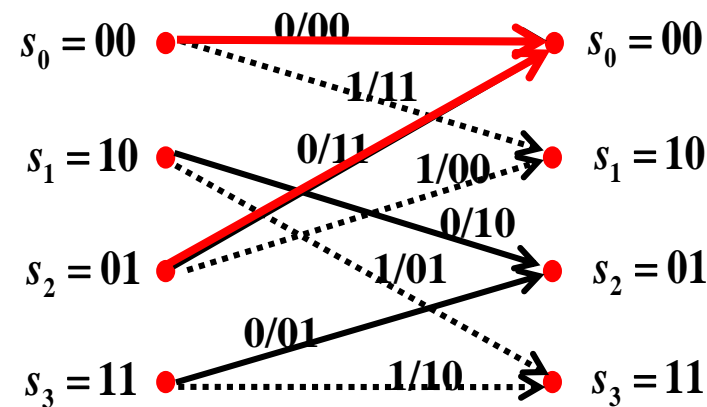
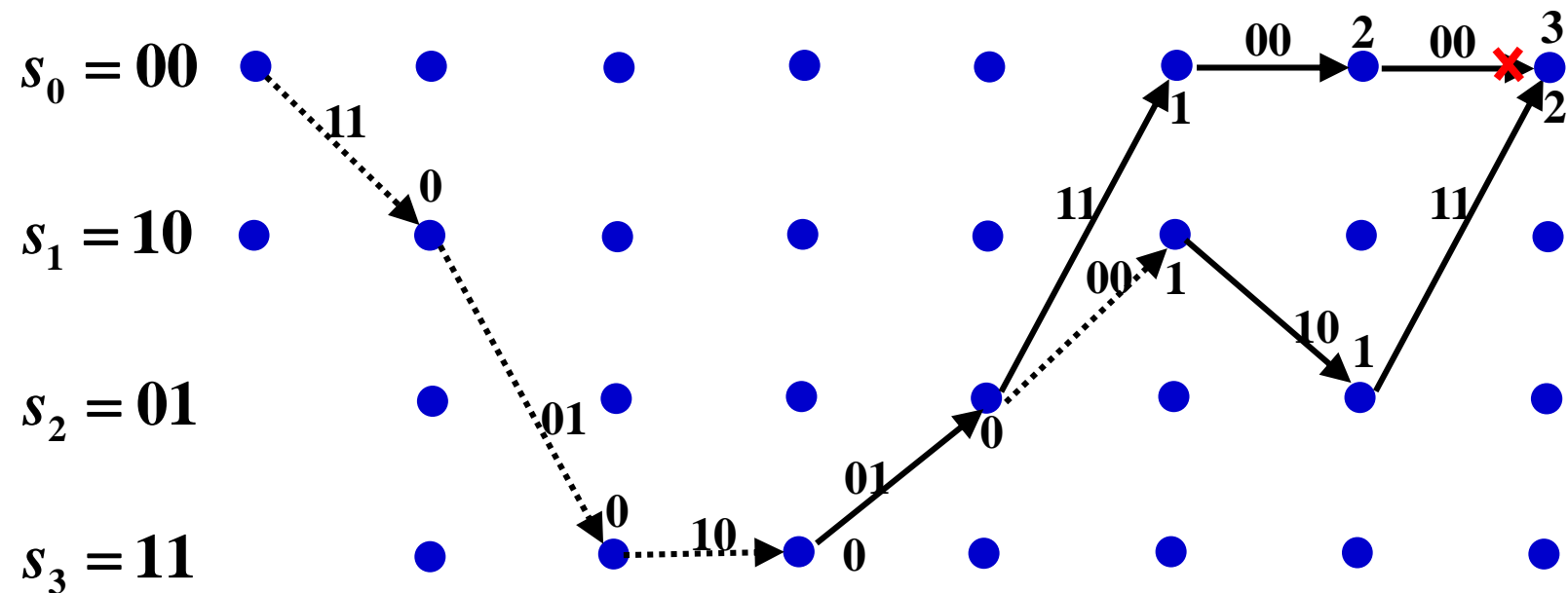
01

01

10

10

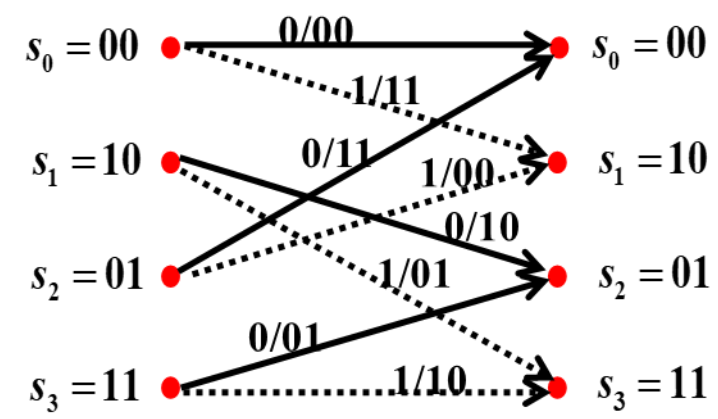
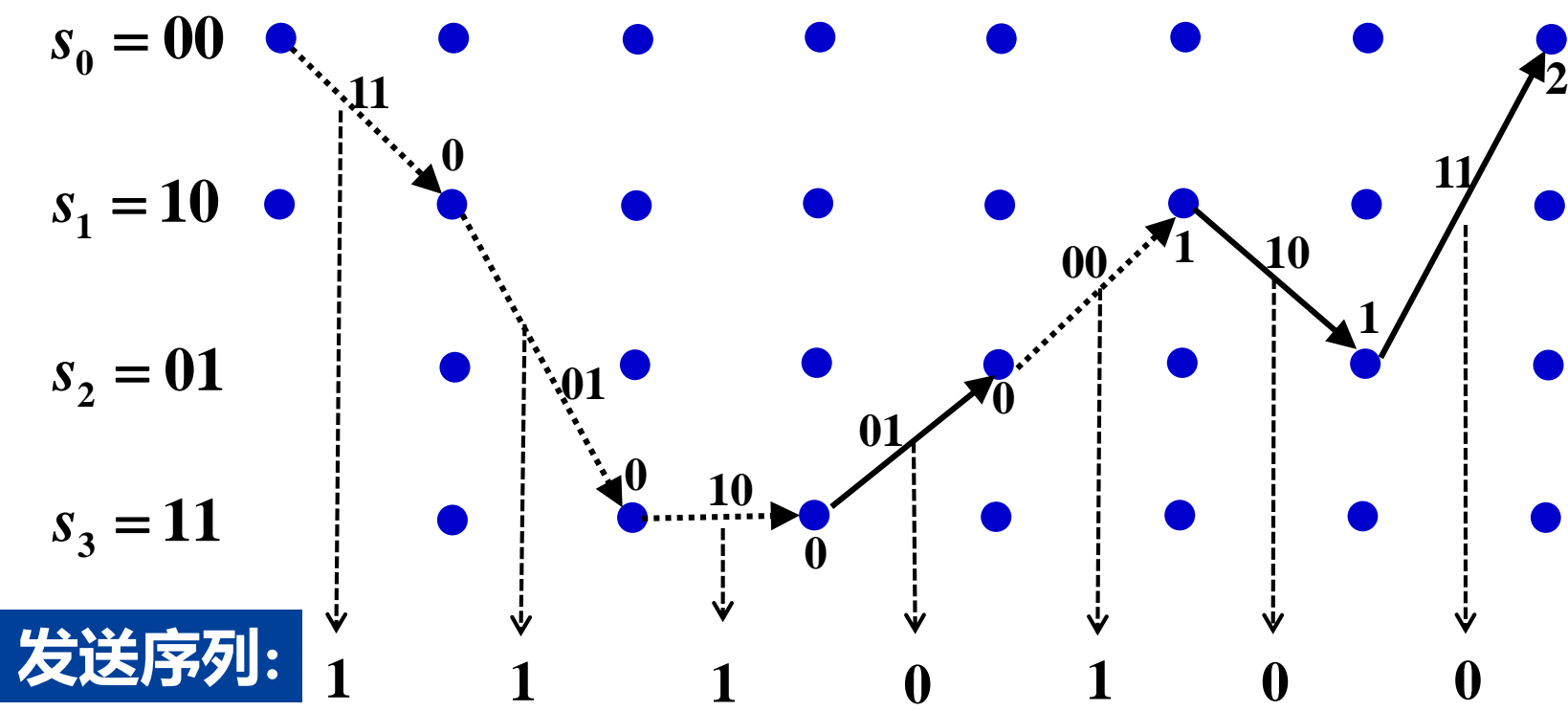
补零



03. 卷积码的译码方法 —— 维特比译码举例（译码）

信息序列M:	1	1	1	0	1	0	0
发送码字C:	11	01	10	01	00	10	11
接收序列R:	11	01	10	01	01	10	10

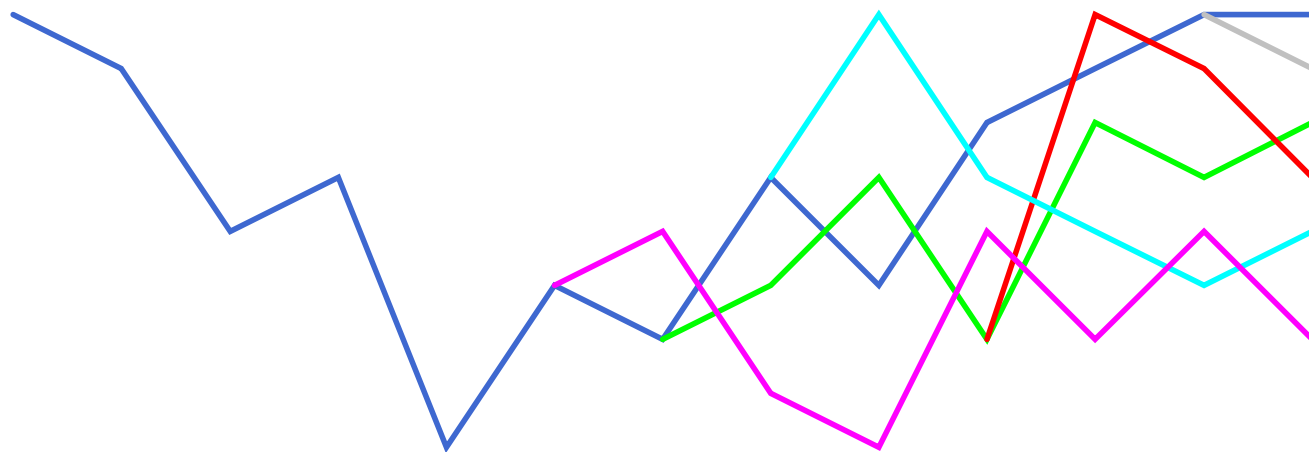
补零



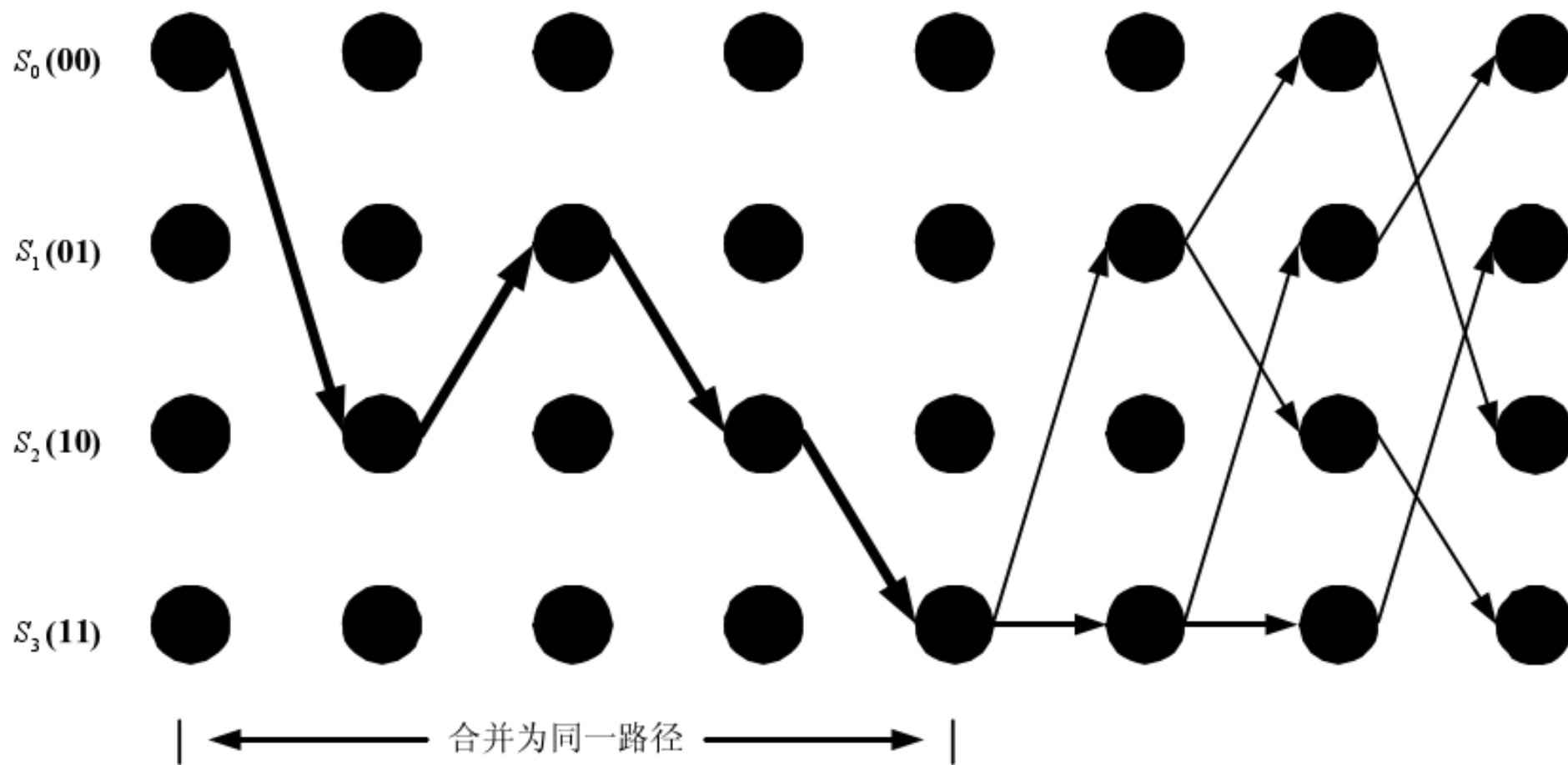
03. 卷积码的译码方法 —— 滑动窗维特比译码算法 (1)

基本思想:

当状态数有限时，给定时刻的各状态残留路径在一定时间 (L) 之前来自于同一状态的可能性随 L 的增加而迅速趋近于1。因此当前时刻各残留路径很可能来自于 L 时刻前的同一路径。



03. 卷积码的译码方法 —— 滑动窗维特比译码算法 (2)



03. 卷积码的译码方法 —— 特点

■ 维特比译码的特点

- 维特比算法是最大似然译码算法
- 运算量和存贮量与状态数 2^{km} 呈线性关系 ((n, k, m) 卷积码)
- 运算量和存贮量与码长 L 呈线性关系

■ 截尾译码

- 分段译码，降低复杂度。
- 译码深度 h ，其中 m 为编码器记忆深度（即约束长度-1）

$$h = (5 - 10)m$$



目 录

01 卷积码的基本概念

02 卷积码的编码方法

• 03 卷积码的译码方法

04 卷积码的性能

05 总结



04. 卷积码的性能 —— 距离特性

- 分组码的纠错能力取决于码字的最小汉明距离。
- 卷积码的纠错能力也取决于码流序列之间的最小汉明距离。
- 通常，人们把序列趋于无穷长时，格图上所有许用路径之间的最小汉明距离叫做**卷积码的自由距离**。
- 自由距离是衡量卷积码性能的主要指标，纠错能力和误码概率都与自由距离直接有关，选用或设计卷积码离不开这个参数。

04. 卷积码的性能——自由距离的确定

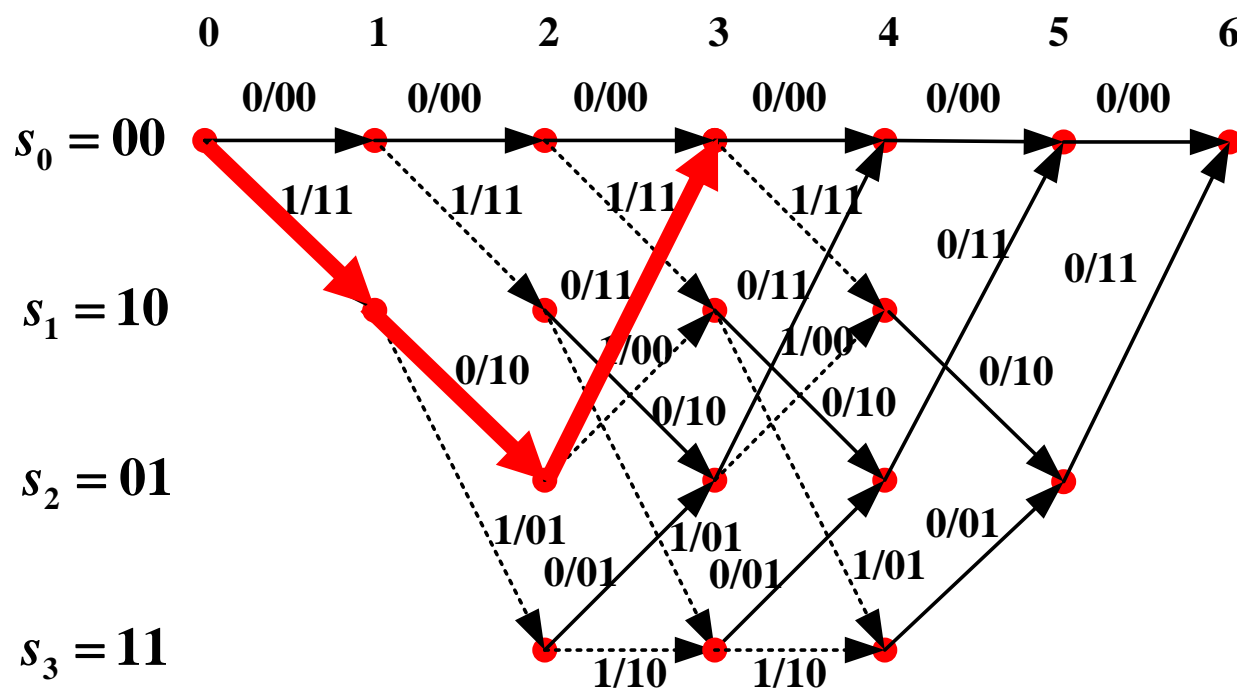
- 卷积码属于线性码，所以任意两个码序列按位模二加之和仍然是一个许用码，而它的重量（1的个数）就等于这两个码序列之间的汉明距离。
- 只要在所有的码序列中找到最小重量的许用码，它的重量就是卷积码的最小汉明距离。而任意许用码序列的重量又等于它与全0码之间的汉明距离。
- 所以，只要在格图上找到一条离全零路径最近的、从0状态出发又回到0状态的非全0路径，那么这条路径所代表的码序列的重量就等于自由距离。

04. 卷积码的性能——自由距 d_f

任意长编码序列之间的最小汉明距

卷积码的自由距等于编码器从全0状态出发，又回到该状态时，所有可能非全0路径重量的最小值。

重量最小的路径：



$$s_0 \xrightarrow{11} s_1 \xrightarrow{10} s_2 \xrightarrow{11} s_0$$

$$d_f = 5$$

在译码深度长度内，可以纠正 $\lfloor (d_f - 1)/2 \rfloor$ 个随机错误。

04. 卷积码的编码增益

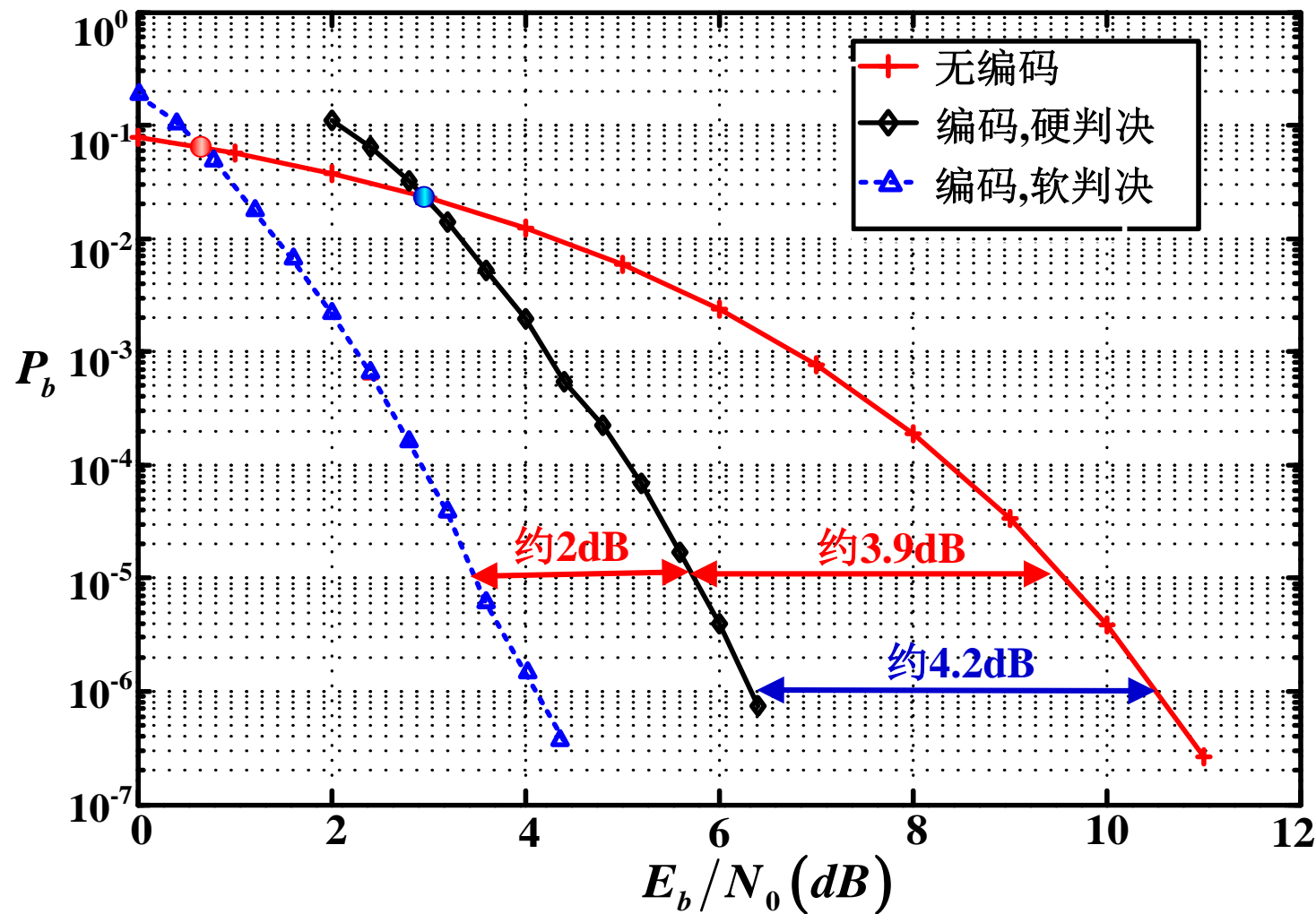
■ 编码增益 (dB) = $[(E_b/N_0)dB]_{\text{未编码}} - [(E_b/N_0)dB]_{\text{编码}}$

■ 硬判决编码增益 (dB) : $10 \log_{10}(R d_f / 2)$

1/2卷积码的自由距和编码增益 (硬判决)

m	g_1 (八进制)	g_2 (八进制)	d_f	编码增益
2	5	7	5	0.97
3	64	74	6	1.76
4	46	72	7	2.43
5	65	57	8	3.01
6	171	133	10	3.98
7	712	476	10	3.98
8	561	753	12	4.77

04. 卷积码的性能 —— 采用 (2,1,8) 卷积码的BPSK系统性能



卷积编码能明显
提升可靠性

04. 卷积码的性能 —— 最常用的卷积码

编码效率	约束长度	自由距离	编码矢量
1/2	3	5	111,101
1/2	4	6	1111,1011
1/2	5	7	10111,11011
1/2	6	8	101111, 110101
1/2	7	10	1001111,1101101
1/2	8	10	10011111,11100101
1/2	9	12	11010111,100011101

短约束长度的最佳卷积码

04. 卷积码的性能 —— 几种重要编码的指标比较

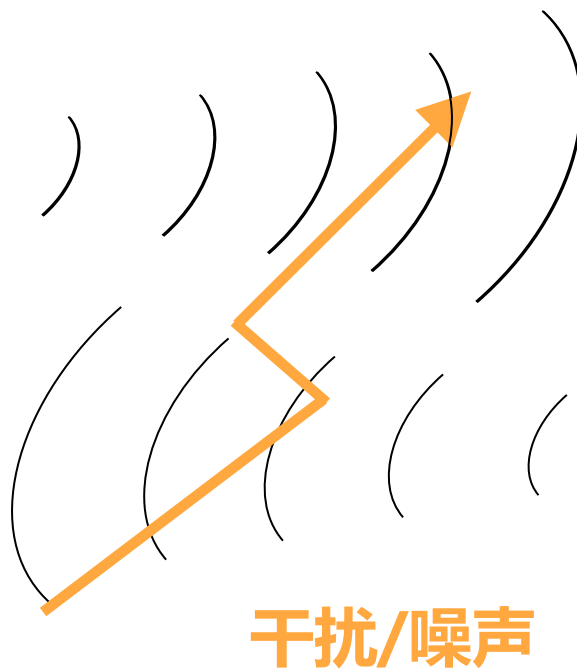
以1/2码率, $P_e < 10^{-5}$ 为例

几种重要编码的指标比较

年份	码型	相对信噪比(dB)
1948	香农码	0
1967	(255,125) BCH码	5.4
1977	卷积码	4.5
1993	Turbo码	0.7
2001	LDPC码	0.0045

04. 卷积码的性能 —— 纠错效果

010101010101



纠错译码后

010101010101





目录

01 卷积码的基本概念

02 卷积码的编码方法

• **03** 卷积码的译码方法

04 卷积码的性能

05 总结



05. 总结

卷积码的编码方法

- (n, k, m) 卷积编码
- 状态转移图
- 网格图

卷积码的译码方法

- 维特比译码是基于动态规划思想的最大似然译码
- 运算量和存储量与码长呈线性关系
- 最重要步骤：加比选

口诀：

沿着网格向前进

每态只留唯一径

对比序列算代价

计算很烦要细心

■ 维特比译码的硬件实现最重要的步骤是加比选，阅读文献后回答：加比选的硬件电路如何实现？

794

IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 18, NO. 5, MAY 2010

Design Space Exploration of Hard-Decision Viterbi Decoding: Algorithm and VLSI Implementation

Irfan Habib, Özgün Paker, *Member, IEEE*, and Sergei Sawitzki, *Member, IEEE*

Abstract—Viterbi algorithm is widely used as a decoding technique for convolutional codes as well as a bit detection method in storage devices. The design space for VLSI implementation of Viterbi decoders is huge, involving choices of throughput, latency, area, and power. Even for a fixed set of parameters like constraint length, encoder polynomials and trace-back depth, the task of designing a Viterbi decoder is quite complex and requires significant

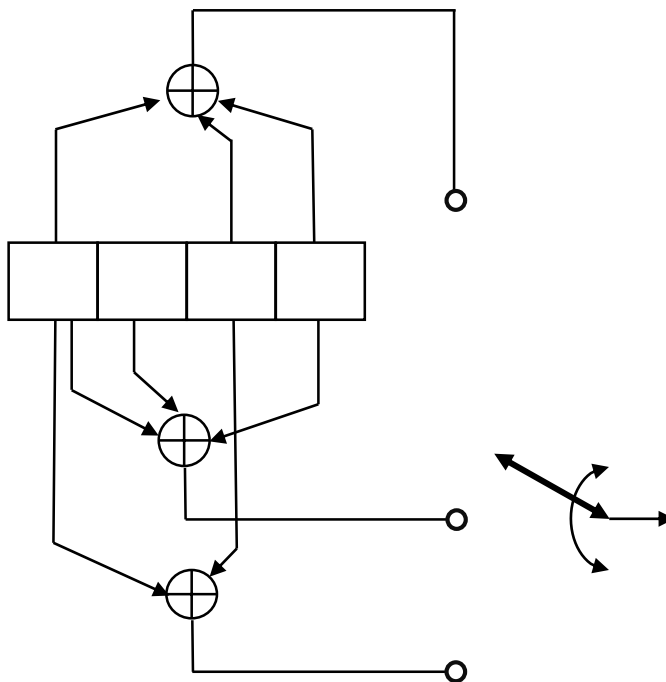
and M- algorithms are not discussed in detail. In addition, almost all of the studied designs optimizations, although proven by synthesis for a specific target CMOS technology, are technology independent, so the conclusions should remain valid for the next two to three solid-state device generations. For this reason, transistor and interconnect level improvements are not



I Habib, O Paker, S Sawitzki. “Design Space Exploration of Hard-Decision Viterbi Decoding: Algorithm and VLSI Implementation”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2010, 18 (5): 794-807.

课后作业

分析图中编码效率为 $2/3$ 的卷积码编码器。在该编码器中，每次有 $k = 2$ 个信息比特移入，同时有 $n = 3$ 比特码元输出。寄存器共有 $kK = 4$ 级，约束长度 $K = 2$ 是指2比特单位个数，编码器的状态定义为最右边的 $K - 1$ 级 k 元组的内容。试画出状态图、树状图和网格图。



谢谢!

