

基础信息论

校验位编码方法

华中科技大学电信学院

校验位编码方法

■ 目的

- 解释校验位编码用于检错/纠错的原理
- 说明如何处理突发错误
- 比较不同信道编码方法的比特率和误码率。

■ 内容

- 基于奇偶校验位编码
- $(9,4,4)$ 码
- 纠正突发错误

基于奇偶校验位编码

单奇偶校验位码

- 给定k比特的信息, 我们可以通过添加1比特来创建 $(k + 1, k, 2)$ 分组码。
- 选择该位以使码字中的 $(k + 1)$ 位之和为偶数。
- 同样, 如果k个消息位的总和为奇数, 则该位为1, 否则为0。
- 该位称为**奇偶校验位**。
- 生成的码字具有偶校验性。

- 示例: $(8,7,2)$ 码

		奇偶校验位
message block:	0 1 1 0 0 1 0	↓
codeword:	0 1 1 0 0 1 0	1
message block:	1 0 1 1 1 0 0	
codeword:	1 0 1 1 1 0 0	0

$$\text{code rate} = \frac{k}{k+1} = \frac{7}{8}$$

检错

- 使用 $(k + 1, k, 2)$ 奇偶校验位码, 我们可以检测到单比特错误
- 如果接收到的码字中1的个数为偶数, 则我们假设没有发生比特错误
- 否则, 我们假设发生了一个比特错误

示例: $(8,7,2)$ 码

消息块:	0 1 1 0 0 1 0
发送码字:	0 1 1 0 0 1 0 1
接收码字:	0 1 1 0 1 1 0 1
接收码字:	0 1 1 1 1 1 0 1

单比特错误 (被检测到)

两比特错误 (没有被检测到)

(8,4,3) 码

- 将消息块排列为2x2矩阵。
- 向每行或每一列添加一个奇偶校验位 (P_i)，以使其具有偶数奇偶校验性。
 - 选择 P_1 ，使得第1行具有偶校验。
 - 选择 P_2 ，使得第2行具有偶校验。
 - 选择 P_3 ，使得第1列具有偶校验。
 - 选择 P_4 ，使得第2列具有偶校验。
- 重新排列这些比特以形成最终的码字。

$$\text{code rate} = \frac{1}{2}$$

D_1 D_2 D_3 D_4 信息块



D_1	D_2	P_1
D_3	D_4	P_2
P_3	P_4	



D_1 D_2 D_3 D_4 P_1 P_2 P_3 P_4 码字

(8,4,3) 码示例

- 将消息块排列为2x2矩阵。
- 向每行或每一列添加一个奇偶校验位 (P_i)，以使其具有偶数奇偶校验性。
 - 选择P1，使得第1行具有偶校验。
 - 选择P2，使得第2行具有偶校验。
 - 选择P3，使得第1列具有偶校验。
 - 选择P4，使得第2列具有偶校验。
- 重新排列这些比特以形成最终的码字。

$$\text{code rate} = \frac{1}{2}$$

0 1 1 1 信息块



0	1	1
1	1	0
1	0	



0 1 1 1 1 0 1 0 码字

校正位

D_1 D_2 D_3 D_4 P_1 P_2 P_3 P_4 接收的码字

↓ 重新排列码字

D_1	D_2	P_1
D_3	D_4	P_2
P_3	P_4	

↓ 计算校正位

D_1	D_2	P_1	S_1
D_3	D_4	P_2	S_2
P_3	P_4		
S_3	S_4		

- 选择 P_i 使得重新排列的码字的每一行或每一列具有偶数个比特。
- 校正位 S_i 在接收到的码字中检查, $S_i = 1$ 表示违反了奇偶校验位 P_i 的条件。

校正位的计算示例

0	1	1	0
1	1	0	0
1	0		
0	0		

0	1	1	0
1	1	1	1
1	0		
0	0		

0	1	1	0
1	0	0	1
1	0		
0	1		

D_1	D_2	P_1	S_1
D_3	D_4	P_2	S_2
P_3	P_4		
S_3	S_4		

如果 $D_1 + D_2 + P_1$ 为偶数

那么 $S_1 = 0$

否则 $S_1 = 1$

如果 $D_3 + D_4 + P_2$ 为偶数

那么 $S_2 = 0$

那么 $S_2 = 1$

如果 $D_1 + D_3 + P_3$ 为偶数

那么 $S_3 = 0$

否则 $S_3 = 1$

如果 $D_2 + D_4 + P_4$ 为偶数

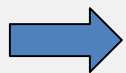
那么 $S_4 = 0$

否则 $S_4 = 1$

执行纠错

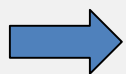


0	1	1	0
1	1	0	0
1	0		
0	0		



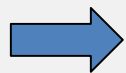
0 1 1 1
修正后的数据
(没有错误)

0	1	1	0
1	1	1	1
1	0		
0	0		



0 1 1 1
 P_2 出错

0	1	1	0
1	0	0	1
1	0		
0	1		

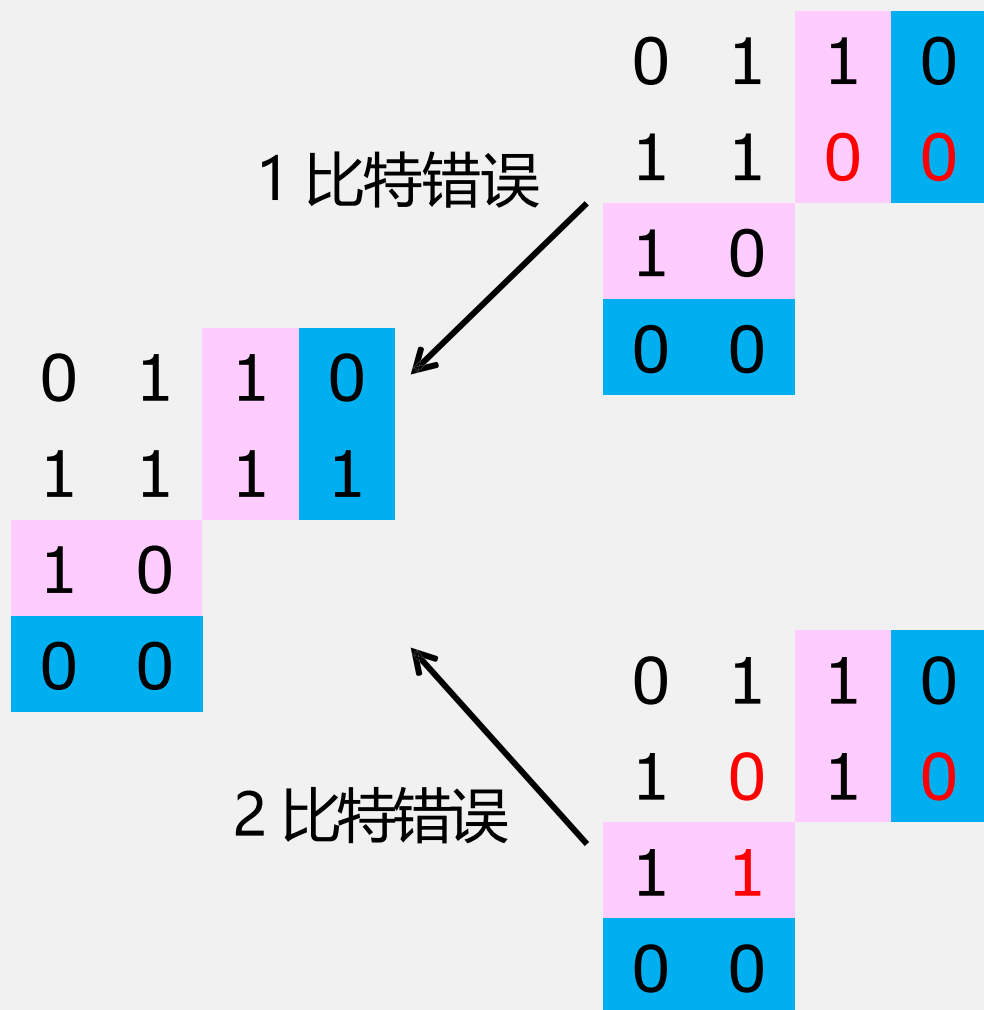


0 1 1 1
 D_4 出错

• 因为 $d=3$, 我们可以检错并纠正 $(d-1)/2=1$ 比特错误.

- 检查综合位
- 如果所有的 $S_i = 0$, 我们就假设没有错误.
- 如果只有一个 $S_i = 1$, 我们假设奇偶校验位 P_i 有错误.
- 如果第 i 列和第 j 行的校正子位为 1, 我们假设位置 i, j 处的数据位有位错误

执行检错



- 对于 (8,4,3)码, 因为 $d=3$, 可以检测 1 或者 2 比特错误.
- 检查校正位
 - 如果所有 $S_i = 0$, 则我们假设没有错误.
 - 否则, 我们假设至少有一个错误

011011101101



0110

1110

1101



01101111

11100101

11010110



01100111

11110101

11000110



0110 1110 1101

1. 取输入的消息流:
2. 将消息流分成k位块 (例如k = 4)
3. 添加 (n-k) 个奇偶校验位以形成n位码字 (例如n = 8)
4. 通过含噪的信道传输数据并接收带有一些错误的码字
5. 执行纠错
6. 从每个校正的码字中提取k = 4个消息位

小结

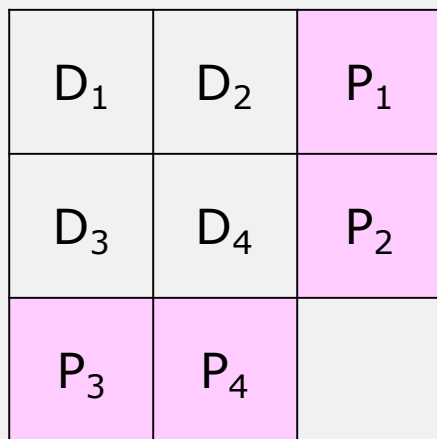
- 通信系统中始终存在的噪声会导致误码。
- 纠错码可用于降低误码率。
- 对于 (n, k, d) 分组码, 我们使用 n 位来编码 k 个消息位, 其中 $n > k$ 。
- 码字之间的最小汉明距离 d 表征该码可以检测或纠正多少位错误。

(9,4,4) 码

回顾: (8,4,3) 码

D_1	D_2	D_3	D_4
-------	-------	-------	-------

 消息块



D_1	D_2	D_3	D_4	P_1	P_2	P_3	P_4
-------	-------	-------	-------	-------	-------	-------	-------

 码字

- 将消息块排列为2x2矩阵。
- 向每行或每一列添加一个奇偶校验位 (P_i)，以使其具有偶数奇偶校验性。
 - 选择 P_1 ，使得第1行具有偶校验。
 - 选择 P_2 ，使得第2行具有偶校验。
 - 选择 P_3 ，使得第1列具有偶校验。
 - 选择 P_4 ，使得第2列具有偶校验。
- 重新排列这些比特以形成最终的码字。
$$\text{code rate} = \frac{1}{2}$$

(9,4,4) 码

D_1	D_2	D_3	D_4
-------	-------	-------	-------

 消息块



D_1	D_2	P_1
D_3	D_4	P_2
P_3	P_4	P_5



D_1	D_2	D_3	D_4	P_1	P_2	P_3	P_4	P_5
-------	-------	-------	-------	-------	-------	-------	-------	-------

- 通过添加整体奇偶校验位，可以将 (8,4,3) 码中的最小汉明距离增加到4，选择该位使得码字始终具有偶校验性。
- (9,4,4) 码使我们可以纠正1比特错误并检测2比特错误
- 或者
- 检测1、2和3比特错误

(9,4,4) 码示例

1	1	1	0
---	---	---	---

 消息块

1	1	0
1	0	1
0	1	1



1	1	1	0	0	1	0	1	1
---	---	---	---	---	---	---	---	---

 码字

- 将消息块排列为2x2矩阵。
- 奇偶校验位
 - P1: D1, D2的奇偶校验
 - P2: D3, D4的奇偶校验
 - P3: D1, D3的奇偶校验
 - P4: D2, D4的奇偶校验
 - P5: D1, D2, D3, D4, P1, P2, P3, P4的奇偶校验
- 重新排列这些位以形成最终的码字。

0	1	1	0
1	1	0	0
1	0	1	
0	0		0

0	1	1	0
1	0	0	1
1	0	1	
0	1		1

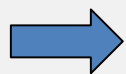
0	1	1	0
1	1	1	1
1	0	1	
0	0		1

D_1	D_2	P_1	S_1
D_3	D_4	P_2	S_2
P_3	P_4	P_5	
S_3	S_4		S_5

- 为了纠正错误，我们假设最多2位有错误。
- 计算5个校正比特位通过检查
 - 偶数奇偶校验 (S_1, S_2) 的前两行
 - 偶数奇偶校验的前两列 (S_3, S_4)
 - 整个码字，以获得偶校验 (S_5)

执行纠错

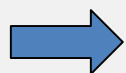
0	1	1	0
1	1	0	0
1	0	1	
0	0		0



0	1	1	1
---	---	---	---

纠正后的数据
(没有错误)

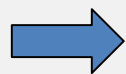
0	1	1	0
1	0	0	1
1	0	1	
0	1		1



0	1	1	1
---	---	---	---

纠正后的数据
(D₄ 出错)

0	1	1	0
1	1	1	1
1	0	1	
0	0		1



0	1	1	1
---	---	---	---

纠正后的数据
(P₂ 出错)

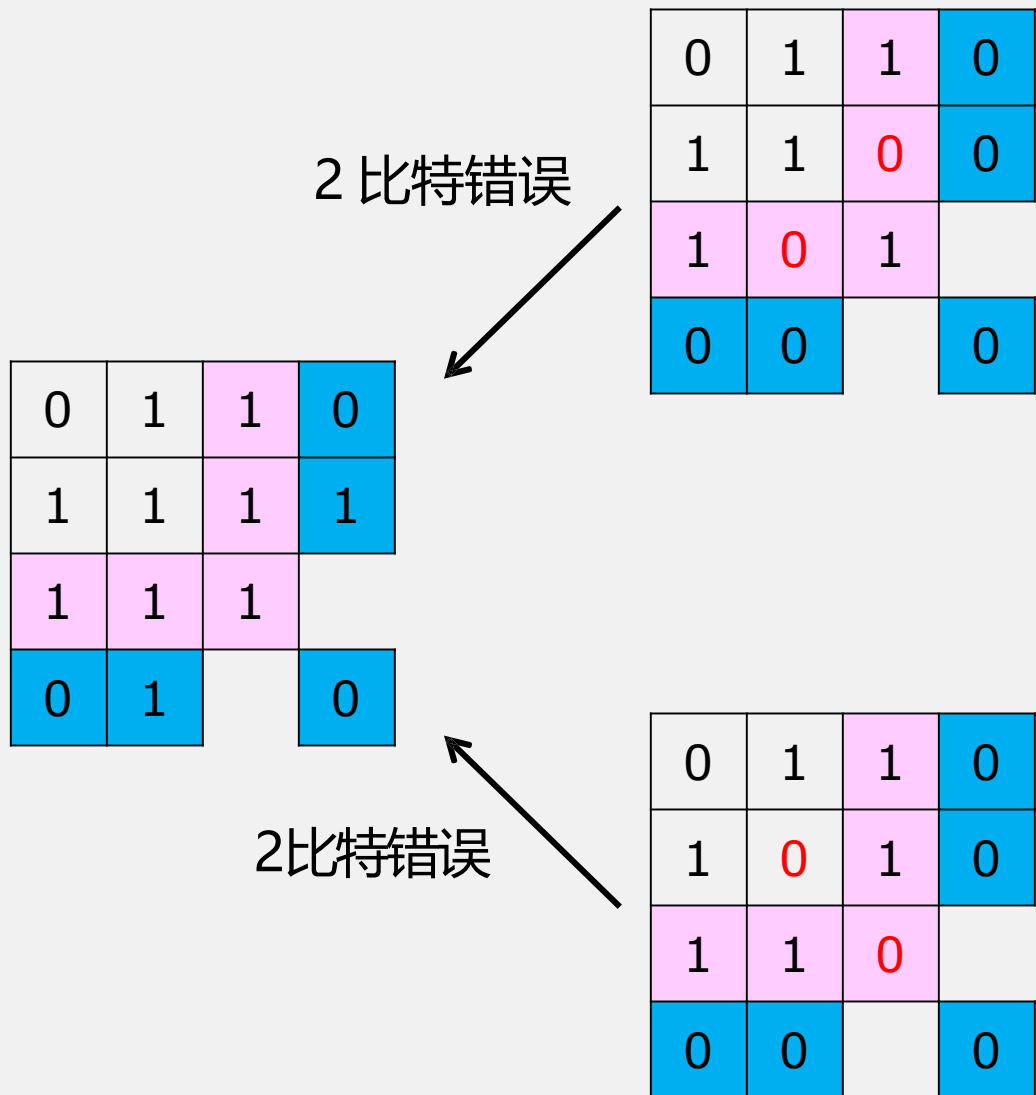
• 检查校正位

如果所有的 $S_i = 0$
没有错误

如果 $S_5 = 1$, % 1比特位出错
检查其他校正位确定是哪1比特出错

% 超过1比特的错误时,
只能检测到错误

检测两比特错误



- 我们无法纠正2比特错误，因为有两个有效码字与接收到的码字相差2比特。

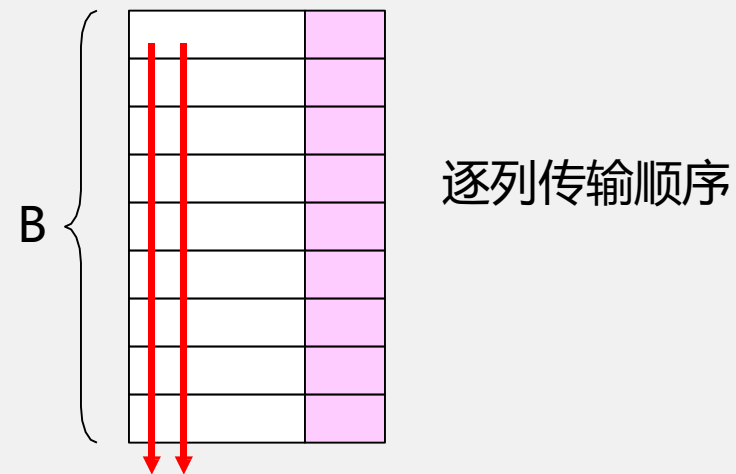
纠正突发错误

突发错误

算法纠正单比特或几个比特的错误性能较好，但是在许多情况下，错误会以多位长的长度突发出现（例如，损坏存储介质，突发无线信道干扰等）。



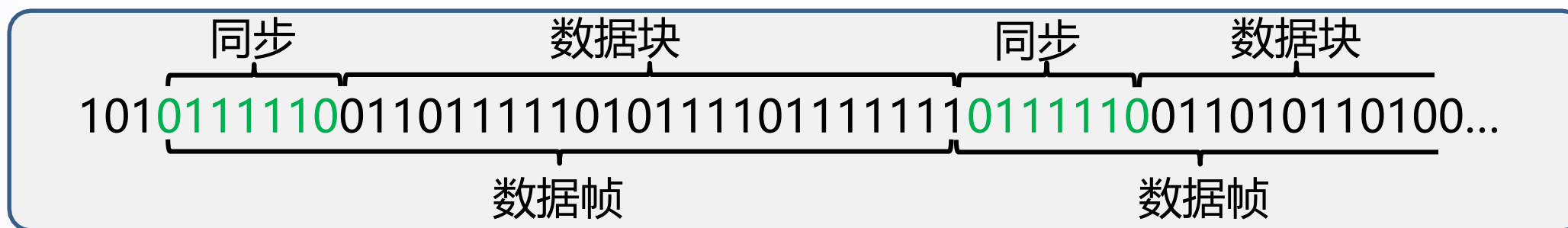
问题：来自特定码字的比特是顺序传输的，因此，B比特块传输会产生多比特错误。



解决方案：**交错传输**来自B个不同码字的比特。现在，B比特块传输会在B个不同的码字中产生1位错误。

数据帧

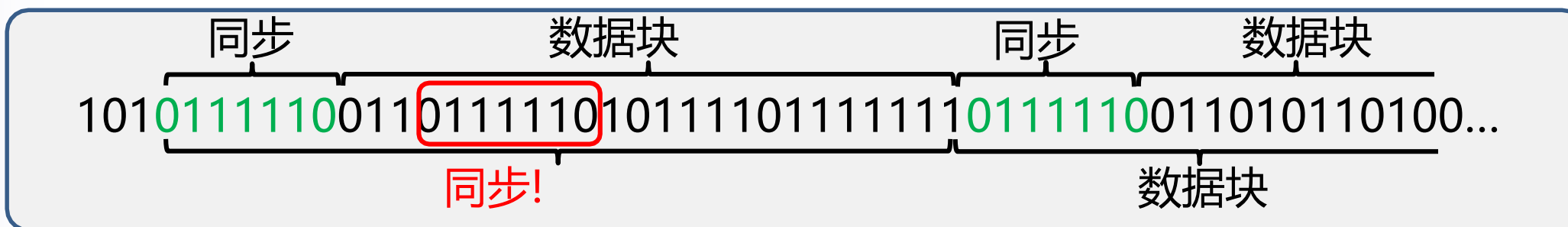
- 查看接收到的比特流，我们如何知道交错码字块从哪里开始？
- 可能的方法
 - 物理指示，例如。
 - 磁盘扇区的开始
 - 单独的控制通道
 - 帧同步序列
 - 设置一个独特的比特模式以标记块的开始（例如起始比特）
 - 同步序列示例：0111110



但是，在添加同步序列之前，我们需要注意一些事项。

比特填充: 动机

- 用于同步序列的比特模式不能出现在帧中的其他位置，否则我们对帧开始的搜索将变得混乱。



- 为了解决这个问题，通过位填充对数据块进行预处理，以确保不出现同步序列。

比特填充

- 如果同步序列为[0111110]，则确保同步序列永远不会出现在数据块中的一种方法是确保连续不会出现五个1。
- 比特填充
 - 扫描数据块
 - 每当发送端发现四个连续的1时，它将0填充（添加）到传出流中
 - 无论下一位是否为1，总是填充0
- 去填充
 - 当接收方看到四个连续的1时，将取消填充（删除）下一位（将为0）。

比特填充示例

输入流



01101111110011110111111111000000

已填充的比特流



0110111101100111100111101111000000

填充比特

去填充的比特流



01101111110011110111111111000000

示例：信道编码

011011101101

获取输入消息

↓
0110
1110
1101

将消息分成k位块($k = 4$)

↓
01101111
11100101
11010110

添加 $(n-k)$ 个奇偶校验位以形成n位码字 ($n = 8$)

↓
011111110001100111101110
↓ ↓ ↓ ↓
01111011100011001111001110

将B个码字的比特交替排列 ($B = 3$)

比特填充

↓
011111001111011100011001111001110

加上同步序列模块 ([0111110])

示例：纠错

011111001111011100011001111001110

发送比特流

bit errors →

011111001111011100100001111001110

收到的比特流（有错误）搜索并删除同步模块

01111011100100001111001110

对数据帧解填充

01111110010000111101110

解交织以形成B个n位码字 (B = 3, n = 8)

01100111
11110101
11000110

执行纠错

0 1 <u>0</u>	1 1 0	1 1 0
1 0 <u>1</u>	1 <u>1</u> 1	0 <u>0</u> 1
1 1	0 1	1 0

提取k = 4个校正的消息比特

0110 1110 1101

作业：本节小测验

登录微助教

<https://www.teachermate.com.cn/>

Thanks

Q & A

Reference

A System View of Communications: From Signals to Packets (Part 1)

<https://www.edx.org/course/system-view-communications-signals-hkustx-elec1200-1x-2>