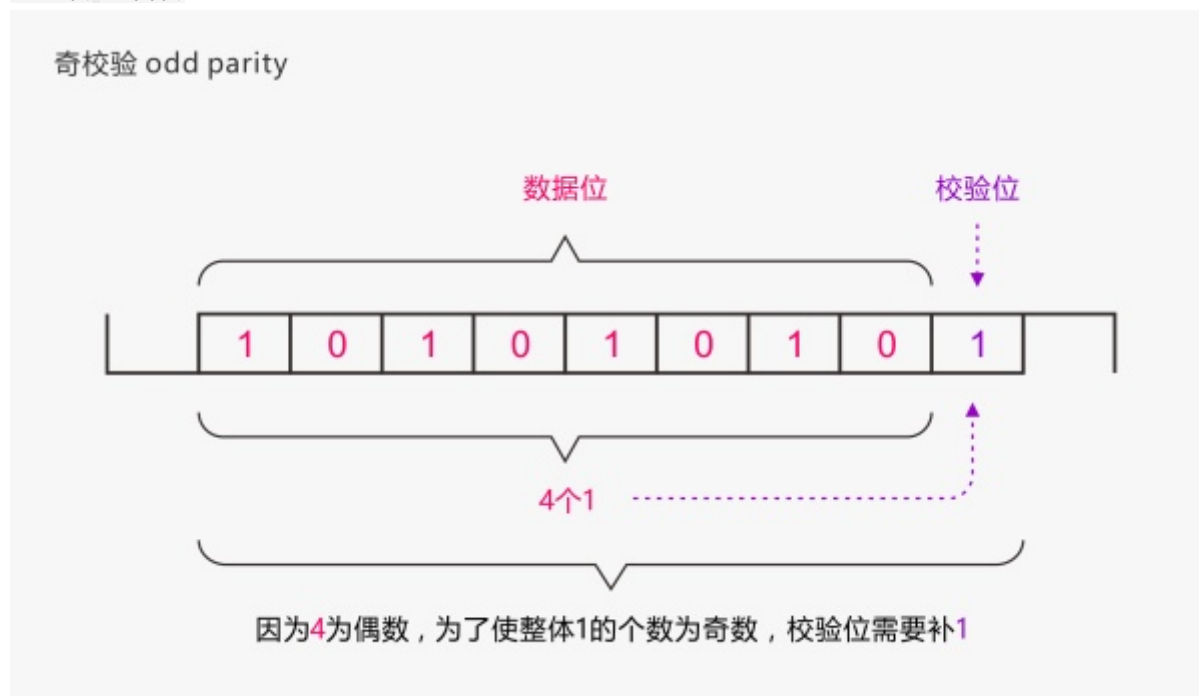


奇偶校验(Parity Check)是一种校验代码传输正确性的方法。根据被传输的一组二进制代码的数位中"1"的个数是奇数或偶数来进行校验。采用奇数的称为奇校验，反之，称为偶校验。采用何种校验是事先规定好的。通常专门设置一个奇偶校验位，用它使这组代码中"1"的个数为奇数或偶数。若用奇校验，则当接收端收到这组代码时，校验"1"的个数是否为奇数，从而确定传输代码的正确性。

奇偶校验需要一位校验位，即使用串口通信的方式2或方式3（8位数据位+1位校验位）。奇校验（odd parity）：让传输的数据（包含校验位）中1的个数为奇数。即：如果传输字节中1的个数是偶数，则校验位为“1”，奇数相反。以发送字符：10101010为例

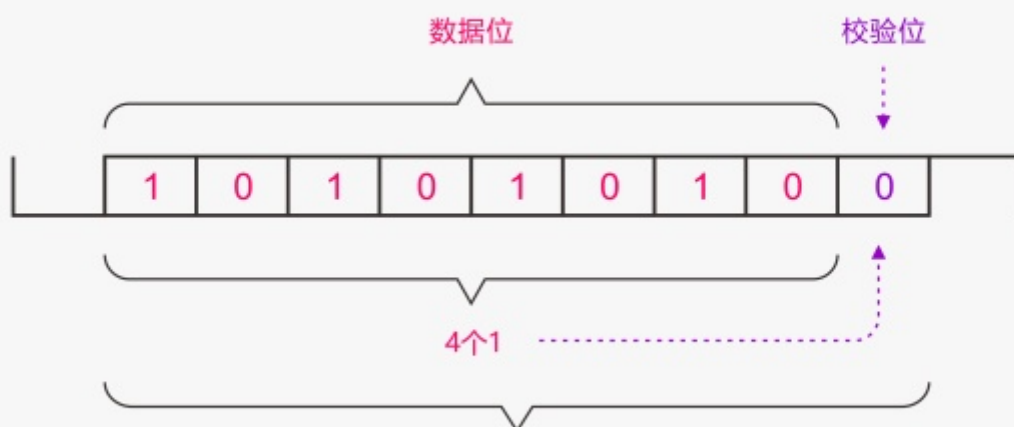
向左转 向右转



偶校验（even parity）：让传输的数据（包含校验位）中1的个数为偶数。即：如果传输字节中1的个数是偶数，则校验位为“0”，奇数相反。还是以发送字符：10101010为例

向左转 向右转

偶校验 even parity



因为4为偶数，为了使整体1的个数为偶数，校验位需要补0

数据和校验位发送给接受方后，接收方再次对数据中1的个数进行计算，如果为奇数则校验通过，表示此次传输过程未发生错误。如果不是奇数，则表示有错误发生，此时接收方可以向发送方发送请求，要求重新发送一遍数据。

优缺点：

- 奇偶校验的检错率只有50%，因为只有奇数个数据位发生变化能检测到，如果偶数个数据位发生变化则无能为力了、(´_ゝ｀)。
- 奇偶校验每传输一个字节都需要加一位校验位，对传输效率影响很大。
- 奇偶校验只能发现错误，但不能纠正错误，也就是说它只能告诉你出错了，但不能告诉你怎么出错了，一旦发现错误，只好重发。
- 虽然奇偶校验有很多缺点，但因为其使用起来十分简单，故目前仍被广泛使用。

应用：

如何用编程确定一个字节中“1”个数的奇偶性？我们可以利用二进制数相加的特点：

$0+0=0$ 、 $1+0=1$ 、 $1+1=0$

可以看出，如果我们将一个字节的各位相加

- 有奇数个“1”的字节的和为1
- 有偶数个“1”的字节的和为0

由此即可通过编程完成判断。实际应用中，实现方法很多，但这是相对简单的一种，这里不再赘述。