五位量子纠错码的各种形式*

陈平形

(国防科技大学应用物理系　长沙　410073)

摘　要　在各种量子纠错方案中五位量子纠错码被讨论得最多, 本文按照正交条件, 用图表对各种错误形式进行分类, 再利用图表的直观性和对称性提供了一个较方便地找出5位量子纠错码各种形式的方法。

关键词　量子计算, 脱散, 纠错

分类号　TP38

Forms of 5-bit Quantum Error Correcting Codes

Chen Pingxing

(Department of Applied Physics, NUDT, Changsha, 410073)

Abstract　In all kinds of quantum error-correcting scheme, five-bit quantum error-correcting codes is discussed most frequently. This paper classifies all kinds of error form with chart according to orthogonality condition, then provides a means to find all forms of five-bit quamtum error-correcting codes conveniently with direct perception and symmetry of chart.

Key words　quantum computation, decoherence, error correction.

由于 Shor 设计了分解大数质因子的具体算法[1], 量子计算机越来越受到人们的重视。然而机器与环境的作用及机器本身各部分之间的作用而引起的脱散将会使输出发生错误[2]。Shor 首先提出了一个方法来消除脱散引起的错误[3], 它是把一个量子位编码成9个量子位(称为9位码)。后来, 又有许多编码方法出现。如7位码[4]; 5位码[5]等。已经证明5位码是位数最小的码(最经济)[5,6]。文献[5]首先给出了一个5位量子纠错码(QECC):

$$0> \to 0_L> = -00000> + 01111> - 10011> + 11100> + 00110> + 01001> + 10101> + 11010> \tag{1}$$

$$1> \to 1_L> = 11111> + 10000> + 01100> - 00011> + 11001> + 10110> + 01010> + 00101> \tag{2}$$

文献[7]证明了如果只有一个量子位出现错误, 则 $i_L>$ 是 $i>$ 的 QECC 的充要条件为:

$$< i_L\ A_k^+ B_{k'}\ j_L > = \lambda_{AB}\delta_{ij} \tag{3}$$

其中 k, k' 取1, 2, 3, 4, 5。i, j 取0, 1。A_k, $B_{k'}$ 是能被 QECC 纠正的分别作用在 k 位和 k' 位的所有可能错误, λ_{AB} 是一个与 $i_L>$ 及 $j_L>$ 有关的复数。(3)式的意义在于: 对所有可能的一位错误, 二个态 $< 0_L$ 、$< 1_L$ 是正交的。

其实5位 QECC 的形式有很多种, 首先不同的编码规则可产生不同的态分量(一个规则产生的态分量我们称之为一个类), 对5位码, 类的个数不多[5]。另外, 也是最为复杂的是, 对同一类各态分量前的正负符号有很多种形式也能产生 QECC, 正因这种复杂性, 文献[5, 6]也只给出少数几种 QECC。找出更多的5位 QECC 能使我们在编码时有更多的自由度, 这有重要的实际意义。本文在第二部分利用(3)式提供一个较简单的方法来找出在一个编码规则下, 各态分类前正负号的所有形式, 以使该态为 QECC。在第三部分讨论了另外编码规则下的5位 QECC。

* 国家部委基金项目资助
1998年4月14日收稿
作者: 陈平形, 男, 1969年生, 讲师

1 一类5位码的所有形式的寻找过程

设某类码的编码规则如下:

$$0> \rightarrow 0_L> = \sum_{r,p,q} r+p, r+p+q, r, p, q>$$

$$1> \rightarrow 1_L> = \sum_{r,p,q} r+p+1, r+p+q, r, p, q> \tag{4}$$

假设上式右边各分量本身带有正负符号。这一部分的目的是求出(4)式各分量的正负号的所有形式, 以使(4)式为 QECC, 即满足正交条件(3)式。

1.1 各种错误形式的分组

把(4)式明显写出为:

$$\begin{aligned} 0_L> = {} & 00000> + 00110> + 01001> + 01111> + 10011> + \\ & 10101> + 11010> + 11100> \\ 1_L> = {} & 10000> + 10110> + 11001> + 11111> + 00011> + \\ & 00101> + 01010> + 01100> \end{aligned} \tag{4′}$$

并按从左到右的顺序把(4′)中 $0_L>$ 的各分量记为1, 2, …, 8, $1_L>$ 中的各分量记为1′, 2′, …, 8′分量。

先以 A, B 分别作用于 $0_L>$ 中第一量子位、$1_L>$ 中第二量子位(简称为1、2位出错)为例来说明分组的方法, 为了方便把(4′)的各分量写成图1的形式, 图1中方括号内的每一行都表示(4′)式中的某个分量, 右边的数字表示分量的序号, 且把出错的位(第1, 2位)写在每分量的左边。下面证明 $0_L>$ 中的第1位和 $1_L>$ 中的第2位分别出现 A、B 错误后正交的充要条件为: 1, 2 1′ 2′这4个分量中其中一个与另三个异号, 且3 4 3′4′, 5 6 5′6′, 7 8 7′8′这三组每组4个分量中任一个也应与另三个异号。

从图中可看出 $0_L>$ 中第1和 $1_L>$ 中第2量子位出错后, $0_L>$ 与 $1_L>$ 的内积为1与1′的内积, 2与2′的内积, …, 8与8′的内积之和。又1与1′的内积等于2与2′的内积, 3与3′的内积等于4与4′的内积, …, 7与7′的内积等于8与8′的内积。例如, 出现 A_1B_2错误(即 A 作用在 $0_L>$ 中第1位, B 作用在 $1_L>$ 中第2位)后1′与1 的内积为:

$$< 00000\ A_1^\dagger B_2\ 10000> = < 0\ A_1^\dagger\ 1> < 0\ B_2\ 0> \equiv a$$

2与2′的内积为:

$$< 00110\ A_1^\dagger B_2\ 10110> = < 0\ A_1^\dagger\ 1> < 0\ B_2\ 0> \equiv a$$

我们把这4个内积分别叫 a、b、c、d, 如果每一组(如1 2 1′2′)中4个分量的任一个(如1分量)与另三个异号, 则每组的二个内积互为负数(如$-a$, a)其和0, 即出错后二态正交。

反之, 如果 $0_L>$ 与 $1_L>$ 正交, 则由于 A, B 形式的任意性, a、b、c、d 都不同, 因此只有 a 与$-a$, b 与$-b$, c 与$-c$, d 与$-d$ 相抵才能和为零, 即必有每组中任一分量与另三分量异号。由这个证明可知, 我们只要把 $0_L>$ 态中出错位相同的二个分量和 $1_L>$ 态中未错位与 $0_L>$ 态的未错位相同的二个分量构成一组, 则只要每组中的一个分量与另三个异号, 就能使出错的态满足正交条件(3)式。因此, 分组的方法是: 把16个分量分成4个组, 每组同态的二个分量出错位相同, 不同态的分量未错位相同。

同理对1、3位, 1、4位, 1、5位, 2、3位, 2、4位, 2、5位, 3、4位, 3、5位, 4、5位出错都可按此特点进行同样分组。通过位与位的交换, 各图中把出错的二位写在每分量的左边, 这样按照上述分组的特点就可清楚看出哪4个分量为一组, 如图2～6。

$$\begin{bmatrix}00&000\\00&110\\01&001\\01&111\\10&011\\10&101\\11&010\\11&100\end{bmatrix}\begin{matrix}1\\2\\3\\4\\5\\6\\7\\8\end{matrix}\qquad\begin{bmatrix}10&000\\10&110\\11&001\\11&111\\00&011\\00&101\\01&010\\01&100\end{bmatrix}\begin{matrix}1'\\2'\\3'\\4'\\5'\\6'\\7'\\8'\end{matrix}$$

图1 第1,2位出错后的分组

Fig.1 Array of 1 bit ,2 bit

$$\begin{bmatrix}00&000\\00&101\\01&010\\01&111\\10&011\\10&110\\11&001\\11&100\end{bmatrix}\begin{matrix}1\\3\\2\\4\\5\\7\\6\\8\end{matrix}\qquad\begin{bmatrix}10&000\\10&101\\11&010\\11&111\\00&011\\00&110\\01&001\\01&100\end{bmatrix}\begin{matrix}1'\\3'\\2'\\4'\\5'\\7'\\6'\\8'\end{matrix}$$

图2 第1,3位出错后的分组

Fig.2 Array of 1 bit, 3 bit

$$\begin{bmatrix}00&000\\00&111\\10&010\\10&101\\01&001\\10&110\\11&011\\11&100\end{bmatrix}\begin{matrix}1\\5\\2\\6\\3\\7\\4\\8\end{matrix}\qquad\begin{bmatrix}11&000\\11&111\\01&010\\01&101\\10&001\\10&110\\00&011\\00&100\end{bmatrix}\begin{matrix}8'\\4'\\7'\\3'\\6'\\2'\\5'\\1'\end{matrix}$$

图3 第2,3位出错后的分组

Fig.3 Array of 2bit,3bit

$$\begin{bmatrix}00&000\\00&111\\10&100\\10&011\\01&001\\01&110\\11&101\\11&010\end{bmatrix}\begin{matrix}1\\6\\2\\5\\3\\8\\4\\7\end{matrix}\qquad\begin{bmatrix}11&000\\11&111\\01&100\\01&011\\10&001\\10&110\\00&101\\00&010\end{bmatrix}\begin{matrix}7'\\4'\\8'\\3'\\5'\\2'\\6'\\1'\end{matrix}$$

图4 第2,4位出错后的分组

Fig.4 Array of 2 bit, 4 bit

$$\begin{bmatrix}00&000\\00&111\\10&010\\10&101\\01&001\\10&110\\11&011\\11&100\end{bmatrix}\begin{matrix}1\\7\\2\\8\\3\\5\\4\\6\end{matrix}\qquad\begin{bmatrix}11&000\\11&111\\01&010\\01&101\\10&001\\10&110\\00&011\\00&100\end{bmatrix}\begin{matrix}6'\\4'\\5'\\3'\\8'\\2'\\7'\\1'\end{matrix}$$

图5 第3,5位出错后的分组

Fig.5 Array of 3 bit,5 bit

$$\begin{bmatrix}00&000\\00&111\\10&100\\10&011\\01&001\\01&110\\11&101\\11&010\end{bmatrix}\begin{matrix}1\\8\\2\\7\\3\\6\\4\\5\end{matrix}\qquad\begin{bmatrix}11&000\\11&111\\01&100\\01&011\\10&001\\10&110\\00&101\\00&010\end{bmatrix}\begin{matrix}5'\\4'\\6'\\3'\\7'\\2'\\8'\\1'\end{matrix}$$

图6 第4,5位出错后的分组

Fig.6 Array of 4 bit, 5 bit

2、5位和3、4位出错后, 不管各分量前的系数如何二态都正交, 另1、2位出错和1、5位出错, 分组完全一样, 1、3位出错和1、4位出错分组完全一样, 故未把它们写出。

1.2 满足正交条件各分量正负号的所有形式

找出同时满足图1、图2、图3、图4、图5、图6(满足某图就是指满足使该图中每组的一个分量与另3个分量异号)的所有形式是我们的目的, 为此, 可分三步(这样分比较方便):第一步分别找出同时满足图1、图2, 同时满足图3、图4, 同时满足图5、图6的所有形式, 第二步找出同时满足图1、图3、图4所有形式和同时满足图1、图5、图6的所有形式, 第三步找出满足图2的第二步的公集, 即同时满足这6个图的所有形式。

第一步。从图中可看出, 要同时满足图1、图2, $1\ 2\ 3\ 4\ 1'\ 2'\ 3'\ 4'$ 这8个分量成为相互制约的一组, 另8个分量$5\ 6\ 7\ 8\ 5'\ 6'\ 7'\ 8'$ 构成一组, 二组相互独立。每组有32种形式, 同时满足图1、图2的形式共32×32种。例如: 在$1\ 2\ 3\ 4\ 1'\ 2'\ 3'\ 4'$ 组中, 若1与$1'$分量同号, 则2与$2'$分量异号, 3与$3'$分量异号, 4与$4'$分量同号。由于同号可同正或同负, 故有16种形式, 若1与$1'$分量异号, 则2与$2'$分量同号, 3与$3'$分量同号, 4与$4'$分量异号,

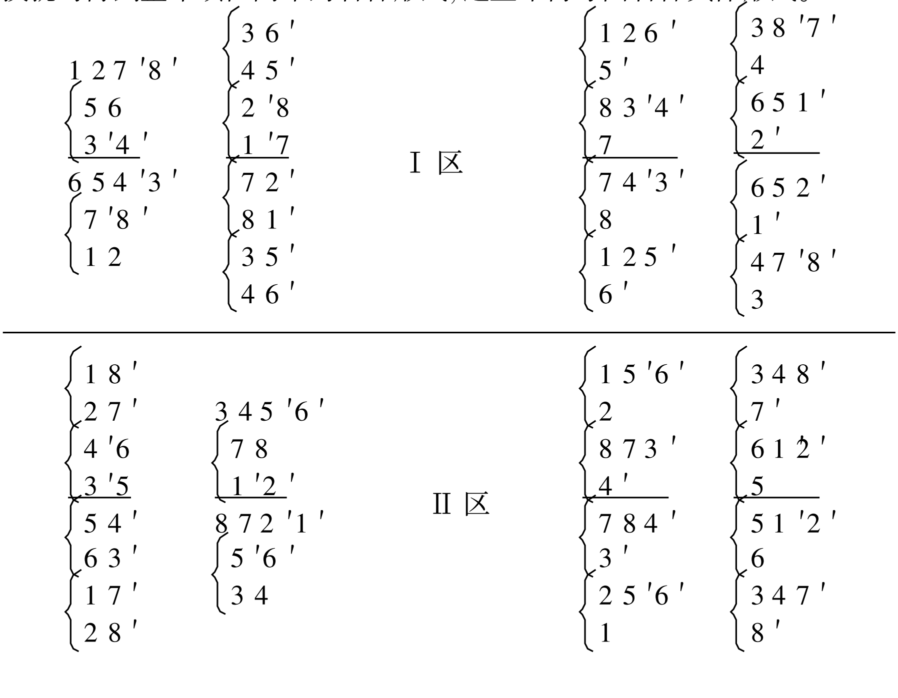
也有16种, 共32种。同理, 同时满足图5、图6时, 1 2 5 6 3 4′ 7′ 8′ 构成一组, 3 4 7 8 1 2′ 5′ 6′ 构成一组, 二组独立, 也有32×32种形式。同时满足图5、图6时, 1 2 7 8 3 4′ 5′ 6′ 构成一组。3 4′ 5′ 6′ 1 2 7 8构成一组, 二组独立有32×32种形式。

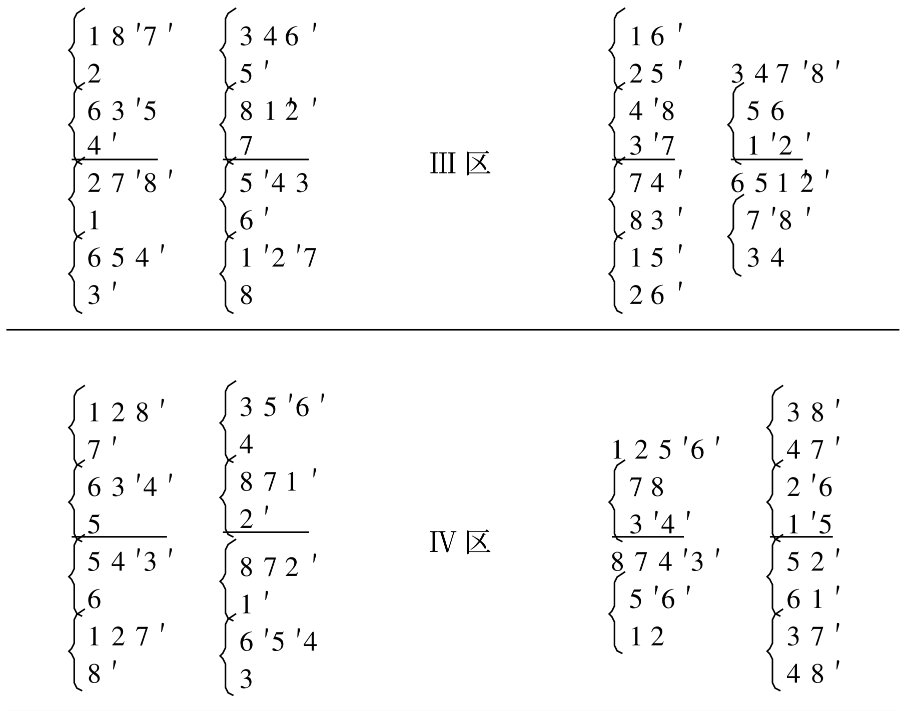
通过一一组合, 不难求出同时满足图3、图4的1 2 5 6 3 4′ 7′ 8′ 组的32种形式, 32种可分为8小类, 每小类4种形式, 如图7(a), 图中每小类用短线分开, 各区间用长线分开, 每小类中同一行的各分量同号, 用大括号相连的二行各分量异号。例如: 7(a)的最上面类中1 2 7 8′ 分量同号, 5 6分量同号, 3 4′ 分量同号, 5 6分量3 4′ 分量异号。同时满足图5、图6的3 4 7 8 1 2′ 5′ 6′ 组的所有形式只需把图7(a)中的1、5、2、6、3′、7′、4、8′ 分别换成3、7、4、8、1′、5′、2′、6′ 即可, 如图7(b)。

同理, 同时满足图5、图6的1 2 7 8 3 4′ 5′ 6′ 组的所有形式可由图7(a)中的7′、8′、5、6分别换成5′、6′、7、8而得到, 如图7(c), 同时满足图5, 图6的3 4 5 6 1 2′ 7′ 8′ 组的所有形式可由图7(a)中的1、2、3′、4′ 换成3、4、1′、2′ 而得到, 如图7(d)。

第二步。把图7(a)中任一类和图7(b)中任一类组合(共64类)都满足图3和图4, 由图1有: 如果1 2分量同(异)号, 则1 2′ 分量异(同)号, 3 4 3 4′ 分量, 5 6 5 6′ 分量, 7 8 7 8′ 分量也有此性质, 满足此性质又满足图3、4的所有形式则只有16类, 即图7(a)中的每一类只能和图7(b)中的二类组合, 把图7(a)和图7(b)用连线分成4个区间, 图7(a)中的任一类只能和同区间的图7(b)中的二类组合。同理, 满足此性质又满足图5、图6的所有形式也只有16类, 即图7(c)中的任一类也只能和同区间的图7(d)中的二类组合。

第三步。不难发现图7中4个小图的公集并无16×16类, 而只有32类, 即图7(a)和图7(b)中任一类只能和同区间的图7(c)和图7(d)中的任一类相容, 与其余的都矛盾, 且这32类都满足图2。由图7中的4个区间, 我们可以很方便地写出我们要求的各种形式, 因为这4个区间及各区间中的小类具有很好的对称性, 首先, 区间Ⅱ、Ⅲ、Ⅳ中的各种形式, 只要将区间Ⅰ中的某些数字变换就可得到, 如只要将区间Ⅰ中的1 2 3 4换成3 4 1 2, 1′ 2′ 3′ 4′ 换成3′ 4′ 1′ 2′ 就可得到区间Ⅲ的各种形式, 另外对某一区间(如Ⅰ区间)中满足图7的各种形式, 只要分别在图7(a)、图7(b), 图7(c), 图7(d)中任取一小类, 求出这4个小类的公集, 然后用适当的数字代换就可得到整个该区间中的各种形式, 这里不再写出各种具体形式。





(a) (b) (c) (d)

图7

Fig. 7

2 别的形式的5位量子纠错码

下面我们讨论别的形式的具有8分量的5位 QECC, 从第二部分可知, 把(4) 式加上上面得到的正负符号组合的任一种都是 QECC。很容易证明: 把(4) 式的各位相互轮换而每个分量前的正负符号不变后所形成的新的形式都是 QECC, 这是因为上面得到的 QECC 是对任一位出错都适合的。

其次, 还有与(4) 式不同的编码规则, 如:

$$0_L> = \sum_{p,q,r} r+p, r+p+q, p, q, r>, \quad 1_L> = \sum_{p,q,r} r+p+1, r+p+q+1, p, q, r>$$

$$0_L> = \sum_{p,q,r} r+q, r+p, r, p, q>, \quad 1_L> = \sum_{p,q,r} r+p+1, r+p, r, p, q>$$

等这些形式各分量的正负号也可仿照前面的步骤求出。

3 结论

通过对各种出错形式的分组, 提出了一种寻找所有形式的5位 QECC 的方法, 这个看似很复杂的工作, 只要能得到图7的形式就可由对称性较方便地完成, 另外从中知道5位 QECC 的形式是很多的, 我们最好是寻找一种能快速地判断某种量子码是否为 QECC 的判据。

参考文献

1 Ekert A, Jozsa R. Quantum computation and shor's factoring algorithm. Rev. Mod. Phys, 1996, 68: 733

2 Wojciech Z H. Decoherence and the transition from quantun to classical. Phys today 1991, 10: 36

3 Shor P W. Scheme for reducing decoherence in quantum computer memory. Rev. A, 1995, 52: 2493

4 Teane A M. Error correcting codes in quantum theory. Phys. Rev. lett, 1996, 77: 793

5 Raymond, Laflamme, et al. Perfect quantum error correting code. Phys. Rev. lett, 1996, 77: 198

6 Chau H F. Five quantum register error correction code for higher spin systems. Phys Rev A 1997, 56: 1

7 Emanuel, Knill, Raymond Lafcamme. Theory of quantum error-correcting codes. Phys. Rev A 1997, 55: 900