第 40 卷第 3 期　　西 南 大 学 学 报（自然科学版）　　2018 年 3 月
Vol. 40 No. 3　　Journal of Southwest University (Natural Science Edition)　　Mar. 2018

DOI：10.13718/j.cnki.xdzk.2018.03.013

定向距离为 d 的 Gray 码[①]

刘　菲[1]，　瞿云云[2]，　包小敏[1]，　黄华伟[2]

1. 西南大学 数学与统计学院，重庆 400715；2. 贵州师范大学 数学科学学院，贵阳 550001

摘要： 本文给出定向距离为 $d(\geqslant 1)$ 的 Gray 码的定义，并给出了两个生成定向距离为 d 的循环 Gray 码的方法.

关　键　词： Gray 码；定向距离；定向距离为 d 的 Gray 码；生成算法

中图分类号：O211.4　　**文献标志码：A**　　**文章编号：** 1673-9868(2018)03-0089-06

Gray 码的编码规则是使任何两个相邻代码只有一个位的状态不同，其余位必须有相同状态. 它的好处在于从某一编码变到下一个编码时只有一位状态发生变化，有利于得到更好的译码波形. Gray 码的类型很多，表 1 列举的是 3 种长度为 3 的 3 进制 Gray 码.

表 1　3 个不同的 3 进制 3 位 Gray 码

第 1 类			第 2 类			第 3 类		
000	120	210	000	122	200	000	210	120
001	121	211	001	121	201	002	212	122
002	122	212	002	120	202	001	211	121
012	102	222	012	110	212	021	201	111
010	100	220	011	111	211	020	200	110
011	101	221	010	112	210	022	202	112
021	111	201	020	102	220	012	222	102
022	112	202	021	101	221	011	221	101
020	110	200	022	100	222	010	220	100

第一种类型的 Gray 码是从全零码字开始，指针在最右边，指针所指的数字加 1，其余数字保持不变. 如果得到的码字与前面的码字不同，则这是新的码字；如果得到的码字与前面的码字相同，则指针左移. 重复上述过程，得到如表 1 所示第 1 类中的 Gray 码. 表 1 所示第 2 类中的 Gray 码，也叫反射 Gray 码[1-3]. 表 1 中第 3 类也是 Gray 码的一种. 这种类型的 Gray 码与表 1 的第 1 类中的 Gray 码，不仅具有 Gray 码的所有特性，而且 Gray 序中后一个码字与前一个码字不同位置处的数字之间有相同的规律：第 1 类中后一个码字减去前一个码字得到的数字模 3 都余 1，这种类型的 Gray 码本文称之为定向距离为 1 的 Gray 码；第 3 类中后一个码字减去前一个码字得到的数字模 3 都余 2，这种类型的 Gray 码本文称之为定向距离为 2 的 Gray 码.

Gray 码除了在通信、硬件设计领域中应用以外，在计算机相关科学的其他方面也有广泛的应用[4-9]. 对 Gray 码进行详细分类，并给出其生成算法，对 Gray 码的研究及更广泛的应用有着积极的作用. 本文给出

① 收稿日期：2016-10-17
基金项目：国家自然科学基金项目(61462016)；贵州省科学技术基金项目(黔科合 J 字[2014]2125 号)；贵州师范大学博士启动项目(0514021)；贵州省教育厅青年科技人才成长项目(黔教合 KY 字[2016]130).
作者简介：刘　菲(1992-)，女，山东东营人，硕士研究生，主要从事编码理论和密码学的研究.
通信作者：包小敏，博士，教授.

定向距离为 d 的循环 Gray 码的定义，并给出了两种生成定向距离为 d 的循环 Gray 码的方法.

本文中 n，m，d 都是正整数，$a \bmod m$ 表示 m 除 a 的余数.

定义1(n 维 m 进制 Gray 码)　所有 m^n 个 n 维 m 进制向量的一个排列称为(n, m)Gray 码，此排列遵循的规则是：任意两个相邻向量只有一个位置上的值不一样. 若最后一个向量和第一个向量也只有一个位置上的值不一样，则称此 Gray 码为(n, m) 循环 Gray 码.

设 m 是一个正整数，任意一个非负整数都可表示成一个 m 进制的数. 将一个 m 进制的数看成一个向量，则有下面的定义.

定义2(自然序)　将整数 $0, 1, \cdots, m^n-1$ 按顺序分别表示成长为 n 的 m 进制向量，就得到所有 m^n 个 n 维 m 进制向量的一个排列，这个排列称为所有 m^n 个 n 维 m 进制向量的自然序.

定义3(定向距离)　对于 $a, b \in Z_m=\{0, 1, \cdots, m-1\}$，定义从 a 到 b 的定向距离为$(b-a) \bmod m$. 对 $\boldsymbol{U}=(a_1, a_2, \cdots, a_n)$，$\boldsymbol{V}=(b_1, b_2, \cdots, b_n) \in {Z_m}^n$，定义从向量 $\boldsymbol{U}$ 到向量 $\boldsymbol{V}$ 的定向距离为 $\sum_{i=1}^{n}(b_i - a_i) \bmod m$.

定义4(定向距离为 d 的 Gray 码)　一个(n, m)Gray 码称为定向距离为 d 的(n, m)Gray 码，简称为(n, m, d)Gray 码，若此 Gray 码中任一向量到后一个向量的定向距离为 d. 若最后一个向量到第一个向量的定向距离也为 d，则称此 Gray 码为(n, m, d) 循环 Gray 码.

对于 m 进制 Gary 码来说，定向距离最大为 $m-1$，最小为1，不可能出现定向距离为0的情形. 下面我们总假设 $d<m$.

文献[5]将二进制 Gray 码的概念推广到 m 进制 Gray 码，给出了一种生成 m 进制 Gray 码的方法；文献[6]给出 m 进制自然序与 m 进制 Gray 码(定向距离为1时)的转换方法，这也是利用自然序来生成定向距离为1的 Gray 码的一种生成方法. 本文对任意正整数 d，考虑定向距离为 d 的 Gray 码.

1　定向距离为 d 的 Gray 码的存在性

首先证明下面的引理.

引理1　集合$\{dt \bmod m \mid t=0, 1, \cdots, m-1\}$包含 m 个元当且仅当 $\gcd(d, m)=1$.

证　当 $\gcd(d, m)=1$ 时，若存在 $0 \leqslant t_1 < t_2 \leqslant m-1$ 使得 $dt_1 \bmod m = dt_2 \bmod m$ 成立，则有 m 整除 $d(t_2-t_1)$，因此 m 整除(t_2-t_1)，但这是不可能的，因为 $0<t_2-t_1<m$.

当 $\gcd(d, m)>1$ 时，设 $m=q\gcd(d, m)$，则 $1 \leqslant q < m$，而 $dq \bmod m=0$. 此时形如

$$dt \bmod m,\ t=0, 1, \cdots, m-1$$

的元中至少有两个相同：0和 $qd \bmod m$.

定理1　当 $\gcd(d, m)>1$ 时不存在(n, m, d)Gary 码.

证　当 $\gcd(d, m)>1$ 时，由引理1知，集合$\{dt \bmod m \mid t=0, 1, \cdots, m-1\}$包含元素的个数少于 m，即从0到 $m-1$ 的数字不可能全部出现. 而(n, m, d)Gray 码若排成一个矩阵，则每一列上的数字都要包含从0到 $m-1$ 这 m 个数字，否则矩阵的行数将少于 q^k. 所以当 $\gcd(d, m)>1$ 时不存在(n, m, d)Gray 码.

下面利用递归的方法来构造(n, m, d) 循环 Gray 码. 由定理1，我们假设 $\gcd(d, m)=1$.

定义 $\boldsymbol{G}(1, m, d)=(0, d, 2d \bmod m, \cdots, (m-1)d \bmod m)^{\mathrm{T}}$，则由引理1知 $\boldsymbol{G}(1, m, d)$ 是$(1, m, d)$ 循环 Gray 码.

对 $0 \leqslant i \leqslant m-1$，定义 $\boldsymbol{G}^{(i)}(1, m, d)$ 如下：

$$\boldsymbol{G}^{(i)}(1, m, d)=((m-i)d \bmod m, (m-i+1)d \bmod m, \cdots, (m-i-1)d \bmod m)^{\mathrm{T}}$$

由于 $\boldsymbol{G}(1, m, d)$ 是$(1, m, d)$ 循环 Gray 码，所以 $\boldsymbol{G}^{(i)}(1, m, d)$ 也是$(1, m, d)$ 循环 Gray 码.

对整数 j，t，记 $\boldsymbol{j}_t=(\overbrace{j, j, \cdots, j})^{\mathrm{T}}$. 令

$$
\boldsymbol{G}(2, m, d)=\begin{pmatrix} 0_m & \boldsymbol{G}^{(0)}(1, m, d) \\ d_m & \boldsymbol{G}^{(1)}(1, m, d) \\ \vdots & \vdots \\ ((m-1)d \bmod m)_m & \boldsymbol{G}^{(m-1)}(1, m, d) \end{pmatrix}
$$

若 $\boldsymbol{G}(2, m, d)$ 中相邻的两行同时出现在子阵 $((dt \bmod m)_m, \boldsymbol{G}^{(t)}(1, m, d))$ 中，其中 $0 \leqslant t \leqslant m-1$，则这两行的第一个分量都为 $dt \bmod m$，而它们的第二个分量则是 $\boldsymbol{G}^{(t)}(1, m, d))$ 中的相邻两行，故这两行的定向距离是 d. 若 $\boldsymbol{G}(2, m, d)$ 中相邻的两行一个是子阵 $((dt \bmod m)_m, \boldsymbol{G}^{(t)}(1, m, d))$ 的最后一行，另一个是子阵 $((d(t+1) \bmod m)_m, \boldsymbol{G}^{(t+1)}(1, m, d))$ 的第一行，则它们的第二个分量都为 $(m-t-1) \bmod m$，而它们的第一个分量分别是 $dt \bmod m$ 和 $d(t+1) \bmod m$，因此这两行的定向距离为 $(d(t+1)-dt) \bmod m=d$，所以 $\boldsymbol{G}(2, m, d)$ 是 $(2, m, d)$ 循环 Gray 码.

对 $0 \leqslant i \leqslant m-1$，定义 $\boldsymbol{G}^{(i)}(2, m, d)$ 如下：

$$
\boldsymbol{G}^{(i)}(2, m, d)=\begin{pmatrix} ((m-i)d \bmod m)_m & \boldsymbol{G}^{(0)}(1, m, d) \\ ((m-i+1)d \bmod m)_m & \boldsymbol{G}^{(1)}(1, m, d) \\ ((m-i+2)d \bmod m)_m & \boldsymbol{G}^{(2)}(1, m, d) \\ \vdots & \vdots \\ ((m-i-1)d \bmod m)_m & \boldsymbol{G}^{(m-1)}(1, m, d) \end{pmatrix}
$$

显然 $\boldsymbol{G}^{(0)}(2, m, d)=\boldsymbol{G}(2, m, d)$. 可以验证 $\boldsymbol{G}^{(i)}(2, m, d)$ 是 $(2, m, d)$ 循环 Gray 码. 令

$$
\boldsymbol{G}(3, m, d)=\begin{pmatrix} 0_{m^2} & \boldsymbol{G}^{(0)}(2, m, d) \\ d_{m^2} & \boldsymbol{G}^{(1)}(2, m, d) \\ \vdots & \vdots \\ ((m-1)d \bmod m)_{m^2} & \boldsymbol{G}^{(m-1)}(2, m, d) \end{pmatrix}
$$

对 $0 \leqslant i \leqslant m-1$，定义 $\boldsymbol{G}^{(i)}(3, m, d)$ 如下：

$$
\boldsymbol{G}^{(i)}(3, m, d)=\begin{pmatrix} ((m-i)d \bmod m)_{m^2} & \boldsymbol{G}^{(0)}(2, m, d) \\ ((m-i+1)d \bmod m)_{m^2} & \boldsymbol{G}^{(1)}(2, m, d) \\ ((m-i+2)d \bmod m)_{m^2} & \boldsymbol{G}^{(2)}(2, m, d) \\ \vdots & \vdots \\ ((m-i-1)d \bmod m)_{m^2} & \boldsymbol{G}^{(m-1)}(2, m, d) \end{pmatrix}
$$

可以验证 $\boldsymbol{G}^{(i)}(3, m, d)$ 是 $(3, m, d)$ 循环 Gray 码.

对 $0 \leqslant i \leqslant m-1$，若 m 个 $(n-1, m, d)$ 循环 Gray 码 $\boldsymbol{G}^{(i)}(n-1, m, d)$ 已经得到，其中 $\boldsymbol{G}(n-1, m, d)=\boldsymbol{G}^{(0)}(n-1, m, d)$，令

$$
\boldsymbol{G}(n, m, d)=\begin{pmatrix} 0_{m^{n-1}} & \boldsymbol{G}^{(0)}(n-1, m, d) \\ d_{m^{n-1}} & \boldsymbol{G}^{(1)}(n-1, m, d) \\ \vdots & \vdots \\ ((m-1)d \bmod m)_{m^{n-1}} & \boldsymbol{G}^{(m-1)}(n-1, m, d) \end{pmatrix}
$$

则同样可以验证 $\boldsymbol{G}(n, m, d)$ 是 (n, m, d) 循环 Gray 码.

例 1　对 $m=3$, $d=2$，按上面的方法得 $(1, 3, 2)$Gray 码和 $\boldsymbol{G}^{(1)}(1, 3, 2)$，$\boldsymbol{G}^{(2)}(1, 3, 2)$：

$$
\boldsymbol{G}^{(0)}(1, 3, 2)=(0\quad 2\quad 1)^{\mathrm{T}}
$$
$$
\boldsymbol{G}^{(1)}(1, 3, 2)=(1\quad 0\quad 2)^{\mathrm{T}}
$$
$$
\boldsymbol{G}^{(2)}(1, 3, 2)=(2\quad 1\quad 0)^{\mathrm{T}}
$$

进而又可得 $(2, 3, 2)$Gray 码如下：

$$\boldsymbol{G}^{(0)}(2,3,2)=\begin{pmatrix}0_3 & \boldsymbol{G}^{(0)}(1,3,2)\\ 2_3 & \boldsymbol{G}^{(1)}(1,3,2)\\ 1_3 & \boldsymbol{G}^{(2)}(1,3,2)\end{pmatrix}=\begin{pmatrix}0&0&0&2&2&2&1&1&1\\0&2&1&1&0&2&2&1&0\end{pmatrix}^{\mathrm{T}}$$

$$\boldsymbol{G}^{(1)}(2,3,2)=\begin{pmatrix}1_3 & \boldsymbol{G}^{(0)}(1,3,2)\\ 0_3 & \boldsymbol{G}^{(1)}(1,3,2)\\ 2_3 & \boldsymbol{G}^{(2)}(1,3,2)\end{pmatrix}=\begin{pmatrix}1&1&1&0&0&0&2&2&2\\0&2&1&1&0&2&1&0&2\end{pmatrix}^{\mathrm{T}}$$

$$\boldsymbol{G}^{(2)}(2,3,2)=\begin{pmatrix}2_3 & \boldsymbol{G}^{(0)}(1,3,2)\\ 1_3 & \boldsymbol{G}^{(1)}(1,3,2)\\ 0_3 & \boldsymbol{G}^{(2)}(1,3,2)\end{pmatrix}=\begin{pmatrix}2&2&2&1&1&1&0&0&0\\0&2&1&1&0&2&2&1&0\end{pmatrix}^{\mathrm{T}}$$

再由 $\boldsymbol{G}(2,3,2)=\boldsymbol{G}^{(0)}(2,3,2)$，$\boldsymbol{G}^{(1)}(2,3,2)$ 和 $\boldsymbol{G}^{(2)}(2,3,2)$ 又可得到表 1 中第 3 类的 $\boldsymbol{G}(3,3,2)$.

文献[6] 中研究的 Gray 码是本文的一个特例：$(n, m, 1)$ 循环 Gray 码.

2 直接生成 $G(n, m, d)$ 的一个算法

第 1 节给出的构造 $\boldsymbol{G}(n, m, d)$ 的方法是个递归的方法，效率较低，在实际应用中不方便. 本节给出一个直接的方法.

算法 1 DirectionalGrayCodeGen(n, m, d)

输入：整数 $m \geqslant 2$，$n \geqslant 1$，$1 \leqslant d < m$ 且 $\gcd(d, m)=1$

输出：$m^n \times n$ 矩阵 $\boldsymbol{G}$

```
    (g_1, g_2, …, g_n) ← (0, 0, …, 0)_{1×n} //1×n 零矩阵
    G ← (g_1, g_2, …, g_n)
    (b_1, b_2, …, b_n) ← (m-1, m-1, …, m-1)_{1×n} // 计数器向量初始化
    p = n // 指针初始化
    while (b_1, b_2, …, b_n) ≠ (0, 0, …, 0)_{1×n} do
      while b_p = 0 and p > 1 do
        p = p - 1 // 指针左移一位
      end while
      g_p ← (g_p + d) (mod m)
      RowAppend (g_1, g_2, …, g_n) toG
      b_p ← b_p - 1 //p 位的计数器减 1
      if p < n then
        (b_{p+1}, …, b_n) ← (m-1, …, m-1)_{1×(n-p)} // 还原 p+1 到 n 位的计数器
        p = n // 指针复位
      end if
    end while
return G
```

定理 2 设 m，n 和 d 都是正整数. 若 $1 \leqslant d < m$，则算法 1 生成 $\boldsymbol{G}(n, m, d)$.

证 当 $n=1$ 时，易知算法 1 生成 $\boldsymbol{G}(1, m, d)$. 当(g_1)初始化为 $\boldsymbol{G}^{(0)}(1, m, d)$ 的最后一个行向量($(m-1)d \bmod m$)时，算法生成 $\boldsymbol{G}^{(1)}(1, m, d)$. 一般地，当$(g_1)$初始化为 $\boldsymbol{G}^{(i)}(1, m, d)$ 的最后一个行向量($(m-i-1)d \bmod m$)时，算法生成 $\boldsymbol{G}^{(i+1)}(1, m, d)$. 设 $n>1$，假设当 $n-1$，且对 $0 \leqslant i < m-1$，$(g_1, \cdots, g_{n-1})$ 初始化为 $\boldsymbol{G}^{(i)}(n-1, m, d)$ 的最后一个行向量时，算法生成 $\boldsymbol{G}^{(i+1)}(n-1, m, d)$.

对输入 n, m, d，随着算法 1 的运行，当计数器向量由初始值$(m-1, m-1, \cdots, m-1)_n$ 顺序变为$(m-1,$

$0,\cdots,0)_{1\times n}$ 时，由归纳假设，算法 1 将生成$(0_{m^{n-1}},\boldsymbol{G}^{(0)}(n-1,m,d))$. 由于$(m-1,0,\cdots,0)_n\neq(0,0,\cdots,0)_{1\times n}$，外部的 while 循环将继续进行. 此时 $p=2$，经过内部的 while 循环后 $p=1$. 此时算法将当前向量，即$(0_{m^{n-1}},\boldsymbol{G}^{(0)}(n-1,m,d))$的最后一个行向量，$(0,g_2,\cdots,g_n)$的第一个分量加 d，生成新的当前向量$(d,g_2,\cdots,g_n)$. 之后计数器向量的第一个分量变为 $m-2$，后面的分量还原，计数器向量更新为$(m-2,m-1,\cdots,m-1)_{1\times n}$. 由于$(g_2,\cdots,g_n)$是 $\boldsymbol{G}^{(0)}(n-1,m,d)$ 的最后一个行向量，在经过外部 while 的 $m^{n-1}-1$ 次循环后算法将生成$(d_{m^{n-1}},\boldsymbol{G}^{(1)}(n-1,m,d))$. 此时的计数器向量为$(m-2,0,\cdots,0)_{1\times n}$. 这样，经过两轮的 m^{n-1} 次 while 循环，算法内部的 $\boldsymbol{G}$ 为

$$\boldsymbol{G}=\begin{pmatrix}0_{m^{n-1}} & \boldsymbol{G}^{(0)}(n-1,m,d)\\ d_{m^{n-1}} & \boldsymbol{G}^{(1)}(n-1,m,d)\end{pmatrix}$$

再经过一次循环，计数器向量更新为$(m-3,m-1,\cdots,m-1)_{1\times n}$，而当前向量为

$$(2d \bmod m,g_2,\cdots,g_{n-1})$$

其中$(g_2,\cdots,g_n)$是 $\boldsymbol{G}^{(1)}(n-1,m,d)$ 的最后一个行向量.

一般地，对 $1\leqslant i\leqslant m-1$，经过 i 轮的 m^{n-1} 次 while 循环，算法内部的 $\boldsymbol{G}$ 为

$$\boldsymbol{G}=\begin{pmatrix}0_{m^{n-1}} & \boldsymbol{G}^{(0)}(n-1,m,d)\\ d_{m^{n-1}} & \boldsymbol{G}^{(1)}(n-1,m,d)\\ \vdots & \vdots\\ ((i-1)d \bmod m)_{m^{n-1}} & \boldsymbol{G}^{(i-1)}(n-1,m,d)\end{pmatrix}$$

再经过一次循环，计数器向量更新为$(m-i-1,m-1,\cdots,m-1)_{1\times n}$，而当前向量为

$$(id \bmod m,g_2,\cdots,g_n)$$

其中$(g_2,\cdots,g_n)$是 $\boldsymbol{G}^{(i-1)}(n-1,m,d)$ 的最后一个行向量. 由归纳假设，经过又一轮 m^{n-1} 次 while 循环，算法将生成$((id \bmod m)_{m^{n-1}},\boldsymbol{G}^{(i)}(n-1,m,d))$. 因此经过 m 轮的 m^{n-1} 次 while 循环，算法将生成

$$\boldsymbol{G}=\begin{pmatrix}0_{m^{n-1}} & \boldsymbol{G}^{(0)}(n-1,m,d)\\ d_{m^{n-1}} & \boldsymbol{G}^{(1)}(n-1,m,d)\\ \vdots & \vdots\\ ((m-1)d \bmod m)_{m^{n-1}} & \boldsymbol{G}^{(m-1)}(n-1,m,d)\end{pmatrix}$$

此时的计数器向量为$(0,0,\cdots,0)_{1\times n}$，满足 while 循环的终止条件，算法输出 $\boldsymbol{G}$，而 $\boldsymbol{G}=\boldsymbol{G}(n,m,d)$.

表 2 是输入 $m=4$，$n=3$，$d=3$ 时的一个例子.

表 2　(3, 4, 3) 循环 Gray 码 $\boldsymbol{G}(3,4,3)$

$b_1b_2b_3$	$g_1g_2g_3$	$b_1b_2b_3$	$g_1g_2g_3$	$b_1b_2b_3$	$g_1g_2g_3$	$b_1b_2b_3$	$g_1g_2g_3$	$b_1b_2b_3$	$g_1g_2g_3$
333	000	302	012	211	330	120	212	023	121
332	003	301	011	210	333	113	202	022	120
331	002	300	010	203	323	112	201	021	123
330	001	233	310	202	322	111	200	020	122
323	031	232	313	201	321	110	203	013	112
322	030	231	312	200	320	103	233	012	111
321	033	230	311	133	220	102	232	011	110
320	032	223	301	132	223	101	231	010	113
313	022	222	300	131	222	100	230	003	103
312	021	221	303	130	221	033	130	002	102
311	020	220	302	123	211	032	133	001	101
310	023	213	332	122	210	031	132	000	100
303	013	212	331	121	213	030	131		

3 结束语

本文提出定向距离为 $d(\geqslant 1)$ 的概念，并给出了两个生成定向距离为 d 的循环Gray码的方法，其中一个是用递归的方法生成 $\boldsymbol{G}(n, m, d)$，但递归的方法效率较低，实际使用起来不够方便，本文又给出直接生成 $\boldsymbol{G}(n, m, d)$ 的算法，对于任意 d，只要满足算法的条件，都能生成定向距离为 d 的Gray码. 对于每一种方法，后面都有详细的例子可供理解.

参考文献：

[1] FLORES I. Reflected Number Systems [J]. IRE Transactions on Electronic Computers, 1992, 5(2): 79－82.

[2] ER M C. On Generating the *N*-ary Reflected Gray Codes [J]. IEEE Transactions on Computers, 2009, 33(8): 739－741.

[3] SANKAR K J, PANDHARIPANDE V M, MOHARIR P S. Generalized Gray Codes [C]. Proceedings of 2004 International Symposium on IEEE, Intelligent Signal Processing and Communication Systems. New York: IEEE Ress, 2004: 654－659.

[4] ZOU J C, LI J F, QI D X. Generalized Gray Code and Its Application in the Scrambling Technology of Digital Images [J]. Applied Mathematics A Journal of Chinese Universities, 2002, 3(3): 363－370.

[5] COHN M. Affine *M*-ary Gray Codes [J]. Inform Control, 1963, 6(1): 70－78.

[6] SHARMA B D, KHANN R K. On *M*-ary Gray Codes [J]. Information Sciences, 1978, 15(1): 31－43.

[7] GUPTA S. Advanced Level Cyclic Gray Codes with Application [J]. International Journal of Electronics Communication and Computer Technology(IJECCT), 2014, 4(3): 619－622.

[8] WANG D, CHEN S M, MA J W. The Interface Technology of Asynchronous FIFOs Based on Gray Code and Its Application [J]. Computer Engineering Science, 2005, 27(1): 58－60.

[9] OUDJIDA A K, TAGZOUT S, BOUZOUIA B, et al. A Reconfigurable Counter Controller for Digital Motion Control Applications [J]. Microelectronics Journal, 1997, 28(6－7): 683－690.

Gray Codes with a Directional Distance of *d*

LIU Fei[1], QU Yun-yun[2], BAO Xiao-min[1], HUANG Hua-wei[2]

1. School of Mathematics and Statistics, Southwest University, Chongqing 400715, China;

2. School of Mathematical Sciences, Guizhou Normal University, Guiyang Guizhou 550001, China

Abstract: The Gray codes with a directional distance of d ($\geqslant 1$) are defined, and two methods for generating cyclic Gray codes with a directional distance of d are presented.

Key words: Gray code; directional distance; Gray code with a directional distance of d; generating algorithm

责任编辑 张 枸