

セグメント型 DA 変換器の魔方陣レイアウト技術による線形性向上

東野 将史*, 小林 春夫 (群馬大学)

Segmented DAC Linearity Improvement With Layout Technique Using Magic Square Masashi Higashino*, Haruo Kobayashi, (Gunma University)

This paper proposes using magic square layout techniques to improve the linearity of a segmented DAC to cancel systematic mismatch effects among unit current (or capacitor) cells. Simulation results and discussions are shown for comparison among magic square, random walk and regular layout techniques.

キーワード：魔方陣, DA 変換回路, 線形性, ミスマッチ, レイアウト
(Magic Square, DAC, Linearity, Mismatch, Layout)

1. はじめに

電子機器は高速化, 小型化が求められており, デジタル回路はそれらに適している。デジタル化の進展に伴い, 多くの電子機器にはデジタルアナログ変換器 (Digital-to-Analog Converter: DAC, DA 変換器) が搭載されている。身の回りの信号(音声, 画像, 光など) はアナログ信号であり, それらを信号処理するにあたり, AD 変換器及び DA 変換器が必要不可欠であり, 高性能なものが求められている。しかし, 半導体素子を構成しているシリコンウェハ上では, MOSFET 特性, R, C 値等にシステムティックな相対ばらつきが存在する (特性, 値がレイアウト配置により傾斜をもつ)。これにより, 入出力信号は線形関係にあるはずが, 素子のミスマッチにより線形性が劣化してしまう問題がある。

そこで本論文では, セグメント型 DA 変換器の線形性の向上を検討した。魔方陣(Magic Square) [1][2]を用いた単位電流セル (または単位容量セル) 配列のレイアウト方法により, 1次及び2次システムティックミスマッチの影響をキャンセルする方法を考案した。魔方陣を用いた場合, 従来の酔歩(Random Walk)を用いた場合, 規則的なレイアウトを用いた場合の比較の数値シミュレーション結果とその考察を示す。

2. セグメント型 DA 変換器の構成と動作

DA 変換器は, バイナリ(Binary) 型とユナリ(Unary) 型があり, 両者を組み合わせたものがセグメント型である (上位ビットをユナリ, 下位ビットをバイナリ型で構成する)。バイナリ型とは, 2進数の要素を加算するものである。ユナリ型は, 2進数データを一旦デコードし, 数値に変換してから単位要素をその個数分加算するものである[6][7]。

〈2・1〉 バイナリ型 DA 変換器

バイナリ型 DA 変換器を図 1 に示す。N ビットのデジタル入力の各ビットを最上位ビット MSB(Most Significant Bit)から順に D_1, D_2, D_N とすると, 出力電圧 V_o は以下の式で表される。

$$V_o = V_{FS} \left(\frac{D_1}{2} + \frac{D_2}{2^2} + \dots + \frac{D_N}{2^N} \right) \quad (1)$$

出力が電流の場合は, 出力電流 I_o は以下の式で表される。

$$I_o = I_{FS} \left(\frac{D_1}{2} + \frac{D_2}{2^2} + \dots + \frac{D_N}{2^N} \right) \quad (2)$$

つまり, 2進重み付けされた電流源もしくは電荷を用意し, この重み付けされた電流源もしくは電荷を, 入力デジタル信号の各ビットで加算するものである。

バイナリ型は素子数が最小であるため, 小型化が可能である。しかし, バイナリ型 DA 変換器はコードの切り替えの際に, グリッチが発生してしまう。また MSB に相当する素子感度が極めて高く素子のミスマッチがあると大きな誤差となってしまう問題がある。特性の単調性も保証されない。

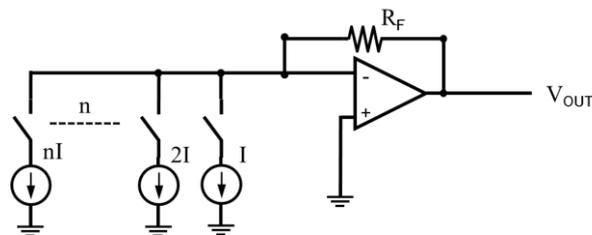


図 1 バイナリ型 DA 変換器

Fig. 1. Binary DAC

<2.2> ユナリ型 DA 変換器

ユナリ型は、最小単位の電圧、電荷もしくは電流を $2^N - 1$ 個用意し、デジタル値に応じて加算することで DA 変換を実現する。図 2 は電流源を用いたものである。デコーダにより、2 進データをデコードし、そのデジタルデータに応じた単位要素分の電流源を ON にすることで、アナログ信号に直すものである。

ユナリ型は、素子のミスマッチがあっても、バイナリ型と比べて出力信号への影響が少ない。グリッチも小さく単調性も原理的に保証される。素子数が多くなってしまうのが欠点である。高線形性の DA 変換器を実現しようとするとき、単位セル（図 2 の単位電流 I）間の相対ミスマッチが問題になり、ここではこの影響を軽減するレイアウト技術を論じる。

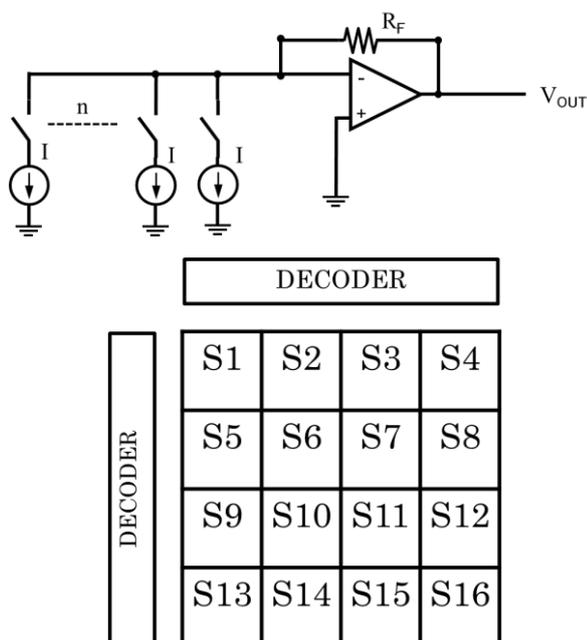


図 2 ユナリ型 DA 変換器と単位電流セルのレイアウト
Fig. 2. Unary DAC circuit and layout of unit cell array.

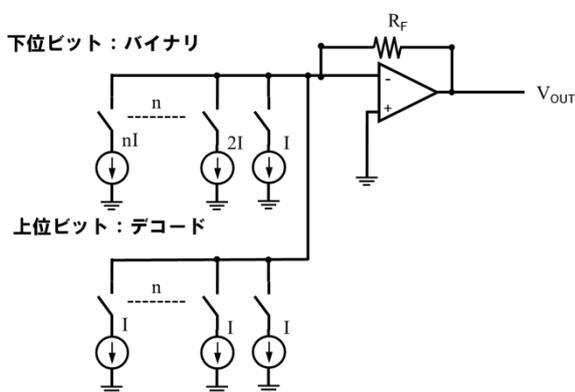


図 3 セグメント型 DA 変換器

Fig. 3. Segmented DAC

<2.3>セグメント型 DA 変換器

多くの DA 変換器では、バイナリ型とユナリ型を組み合わせたものが用いられている。上位ビットには素子感度の

低いユナリ型を用い、下位ビットでは、素子数の少ないバイナリ型が用いられる。これにより、高精度な DA 変換器を適正な回路規模・消費電力で実現できる。

3. 素子のばらつき

回路を構成している素子は IC チップ上で特性の製造ばらつきが生じる。それらのばらつきは配置（場所）によるシステムティックなばらつきと、配置に依存しないランダムなばらつきが存在する。これにより、DA 変換器の入力信号と出力信号は、本来であれば線形関係にあるはずであるが、非線形になってしまう問題がある。各ばらつきの原因について示す[2][3][5][7]。

1) システムティックなばらつき

- ・配線での電圧降下
- ・温度分布
- ・CMOS の製造プロセス
 - a) ドーピング分布
 - b) 酸化膜の厚さによる、しきい値電圧の変化
- ・ウエハ面内の精度
- ・機械的ストレス

2) ランダムなばらつき

- ・デバイスのミスマッチ

システムティックなばらつきは、配置による一次の二次の傾斜のばらつきがある。回路上では、それらを足し合わせたものが、実際のばらつきとなって回路動作に影響してくる。一次及び二次のばらつきに関して以下に示した。

1) 一次の傾斜のばらつき (Linear gradient)

- ・配線での電圧降下
- ・CMOS の製造プロセス

2) 二次の傾斜のばらつき (Quadratic gradient)

- ・温度分布
- ・ウエハ面内の精度
- ・機械的ストレス

上記のばらつきは、セグメント型 DA 変換器の入出力信号の線形性に大きく影響する。DA 変換器を構成する電流源のばらつきを図 4, 5, 6 に示した。(x, y) をその素子のチップ上での位置の座標とすると一次及び二次のばらつきは、以下の式で表される。

1) 一次

$$\varepsilon_l(x, y) = g_l \cdot \cos \theta \cdot x + g_l \cdot \sin \theta \cdot y \quad (3)$$

θ : 傾きの角度, g_l : 傾きの大きさ

2) 二次

$$\varepsilon_q(x, y) = g_q \cdot (x^2 + y^2) - a_0 \quad (4)$$

g_q : 変化量, a_0 : 位置

3) 一次+二次

$$\varepsilon_j(x, y) = \varepsilon_l(x, y) + \varepsilon_q(x, y) \quad (5)$$

一次、二次及び、一次+二次のばらつきを図 4, 5, 6 に示す。

システムティックなばらつきの DA 変換器の線形性への影響は、回路のレイアウト技術により緩和できることが知られている。セグメント型 DA 変換器の場合、従来方法であ

る酔歩(random walk)によりばらつきを軽減し、線形性を向上させている。

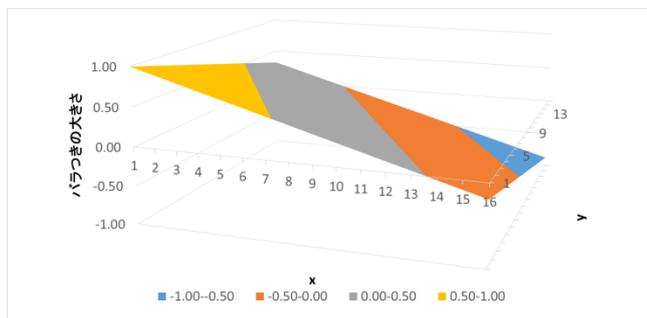


図4 一次のばらつき
Fig. 4. Linear Error

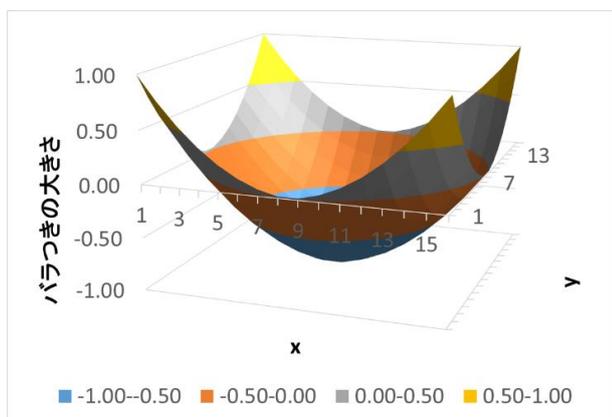


図5 二次のばらつき
Fig. 5. Quadratic Error

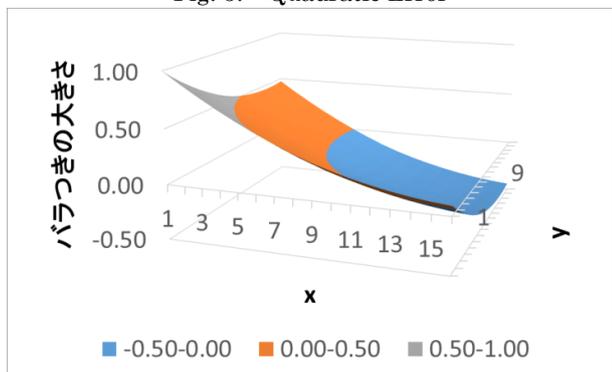


図6 一次+二次のばらつき
Fig. 6. Joint Error

4. 魔方陣について

魔方陣には、各行・列・対角成分の要素の和が全て一定である性質をもつ。この性質から、ユナリ型 DA 変換器の単位セルの配列のバランスがよいのではと考え、DA 変換器におけるレイアウトに応用できる（システムティックの影響を軽減できる）のではないかと考えた。魔方陣の性質及び作り方について説明する[1][2]。

〈4.1〉 魔方陣の性質

魔方陣とは、1 から始まる連続した異なる自然数を $n \times n$ の

基盤の目状に並べ、各行、列及び対角線上の数の和が全て等しいものである。この性質を定和性と呼ぶ。各行、列、対角線上に含まれる数が n 個であることから、一般 $n \times n$ である魔方陣を n 次の魔方陣あるいは n 次方陣と呼ぶ。また、 n 次方陣の各行、列、対角線要素の定和 S は、以下の式で表される。

$$S = \frac{n^2(n^2+1)}{2} \quad (6)$$

図 7 に示した魔方陣は、各行・列・対角成分の要素の和が全て一致していることが確認できる。

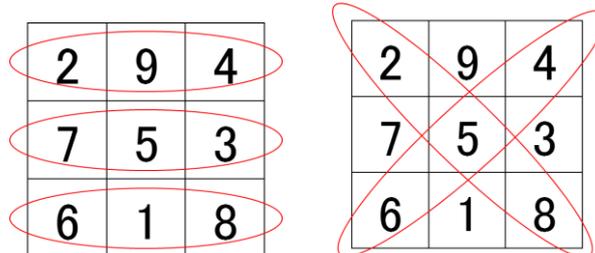


図 7 定和性の一致

Fig. 7. Equivalent constant sum characteristics.

回転移動及び裏返しを行っても、定和性は満たされる。

1) 回転移動

図 8 の(1)の 3 次方陣の中心の周りを左に 90°回転すると(2)が得られる。(3)の方陣を右に 90°回転させると(4)の方陣が得られる。つまり、魔方陣は、方陣を回転させても、定和性が失われない。この性質を回転移動と呼ぶ。

2) 裏返し

図 8 の上段にある魔方陣を中央に対して左右対称にしたものが下段の方陣である。このように、中央軸に対して左右対称に移動させることを裏返しと言う。裏返しを行っても、定和性は失われない。

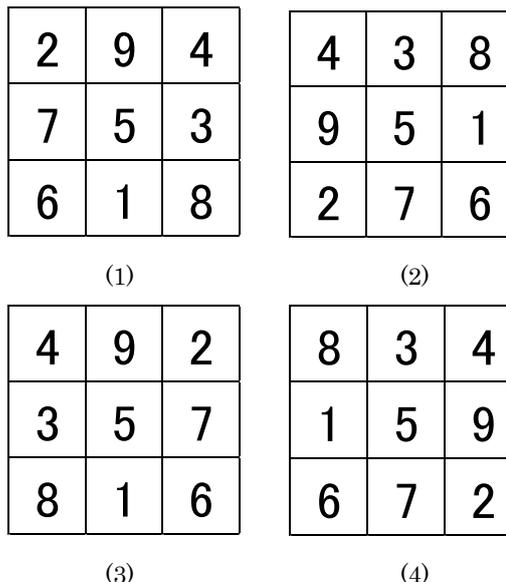


図 8 魔方陣と定和性の保持

Fig. 8. Magic squares and constant sum.

これら性質を利用して、電流源を用いたセグメント型 DA 変換器のスイッチング順序に魔方陣を利用することによって、システムティックなばらつきを軽減できるのではないかと考え、魔方陣による単位セルレイアウトのアルゴリズムを考察した。そのアルゴリズム及び結果、考察を 5 章以降に示した。

〈4・2〉 魔方陣の作り方

5 次方陣を例に一般的な魔方陣の作り方を示す。

n 次方陣を作るとき、n 進法を用いる。ここでは、5 次方陣を作るので 5 進法を用いる。

5 進法で 1~25 を表すと

1 = 01, 2 = 02, 3 = 03, 4 = 04, 5 = 05, 6 = 10, …

…, 21 = 41, 22 = 42, 23 = 43, 24 = 44, 25 = 100

25 だけが 3 桁なので、各数から 1 を引き 00~44 までの 2 桁で、1~25 を表す。そして、0~4 の数を用いて上位桁と下位桁の二種類の方陣をつくる。これを補助方陣と呼ぶ。

図 9 に上位桁と下位桁の補助方陣を示した。

4	2	3	0	1	4	3	0	2	1
1	3	0	4	2	2	1	3	0	4
0	1	2	3	4	1	0	2	4	3
2	0	4	1	3	0	4	1	3	2
3	4	1	2	0	3	2	4	1	0

上位桁

下位桁

図 9 補助方陣の作成

Fig. 9. Supporting magic squares

これらの補助方陣を重ね合わせ、全ての増すに 1 を足す事により、5 次方陣が完成する。ただし、補助方陣同士の数字が 2 度以上重ならないようにしなければならない。この数字が重ならない関係を直交関係と言う。完成した 5 次方陣を図 10 に示した。

25	14	16	3	7
8	17	4	21	15
2	6	13	20	24
11	5	22	9	18
19	23	10	12	1

図 10 5 次方陣

Fig. 10. Fifth-order magic square

〈4・3〉 魔方陣の種類・特徴

魔方陣には様々な種類が存在する。以下解析で利用した魔方陣を示す。用いた魔方陣は同心魔方陣及び対称魔方陣である。また、魔方陣によっては、ある特別な特徴を持つ

ているものも存在する。

1) 同心魔方陣

同心魔方陣は、魔方陣の外側からひと側ずつ取り除いていっても、残る部分が常に定和性を失わないものである。解析で用いた 8 次の同心魔方陣を図 11 に示す。この方陣を 4 つ組み合わせることにより、8bit のセグメント型 DA 変換器を実現する。

59	5	4	62	63	1	8	58
9	18	17	49	50	42	19	56
55	20	28	33	29	40	45	10
54	44	38	31	35	26	21	11
12	43	39	30	34	27	22	53
13	24	25	36	32	37	41	52
51	46	48	16	15	23	47	14
7	60	61	3	2	64	57	6

図 11 8 次同心魔方陣

Fig. 11. Eighth-order concentric magic square

2) 対称魔方陣

対称魔方陣は、中心に関して対象の位置にある 2 数の和が全て $n^2 + 1$ になっている方陣である。図 12 に解析で用いた 8 次対称魔方陣を示した。この対称魔方陣は以下のような性質がある。

- 16 に分割した 16 個の 2×2 の小正方形に含まれる 4 数の和は、全て 130 である。
- 8 次方陣を 16 個のサブ方陣に分解するとき、各サブ方陣に含まれる数の和は全て 130 である。対角線上を除いて各行列は定和 130 を満たす。

1	63	62	4	5	59	58	8
56	10	11	53	52	14	15	49
48	18	19	45	44	22	23	41
25	39	38	28	29	35	34	32
33	31	30	36	37	27	26	40
24	42	43	21	20	46	47	17
16	50	51	13	12	54	55	9
57	7	6	60	61	3	2	64

図 12 8 次対称魔方陣

Fig. 12. Eighth-order symmetric magic square

5. 同心魔方陣による解析と考察

〈5・1〉 同心魔方陣を用いたアルゴリズム

図 11 に示した同心魔方陣を 4 つ組み合わせ、図 12 の

ような 8bit セグメント型 DA 変換器のレイアウトを考案した。ただし、A は図 11 の魔方陣で、B は左に 45° 回転させたものである。

スイッチングする順序は以下の通りである。

1. A1 の 1
2. A2 の 1
3. B1 の 1
4. B2 の 1
5. A1 の 2
- ⋮
255. B1 の 256
256. B2 の 256

規則的にスイッチングした場合は、左上の隅の 59 から右隅へ順にスイッチングしていき、6 までスイッチングした後、次の行の 9 から 57 までスイッチングし、最後の A2 の 6 まで順にスイッチングを行う。

酔歩 (Random Walk) のアルゴリズムは、[4] に従った。

ばらつきは、式(3)~(5)を用いた。一次、二次、一次+二次のばらつきが 3 つの手法によりどう変化するか解析した。

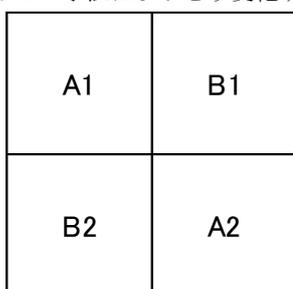


図 13 8bit 同心魔方陣を用いたアルゴリズム

Fig. 13. Layout Algorithm using 8-bit eight-order concentric magic square

〈5・2〉 解析結果と考察

1) 一次のばらつき

最大値は、 ± 1 になるように設定し、角度 θ のみを変化させて、そのときの INL (積分非直線性) の変化を調べた。

● $\theta = 30^\circ$

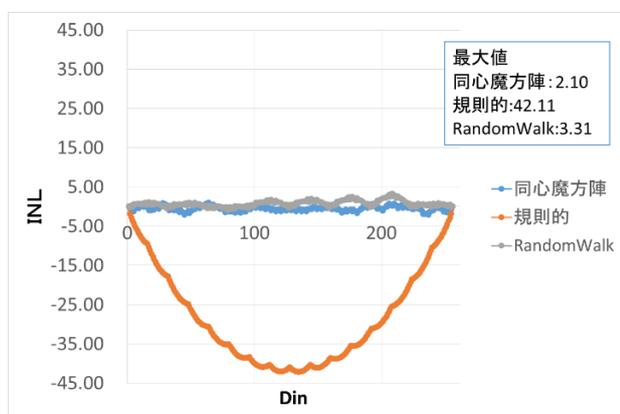


図 14 $\theta = 30^\circ$ の時の INL シミュレーション結果

Fig. 14. Simulated INL ($\theta = 30^\circ$)

● $\theta = 45^\circ$

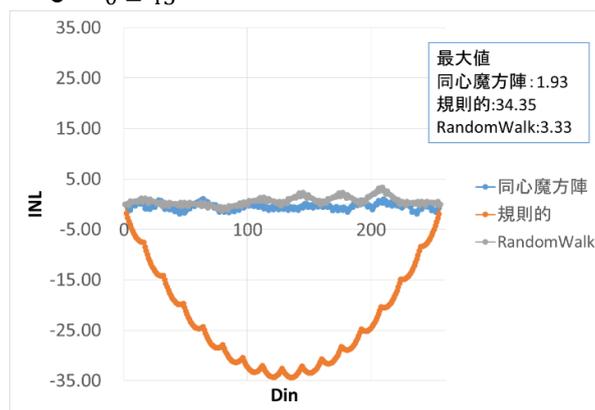


図 15 $\theta = 45^\circ$ の時の INL シミュレーション結果

Fig. 15. Simulated INL ($\theta = 45^\circ$)

● $\theta = 60^\circ$

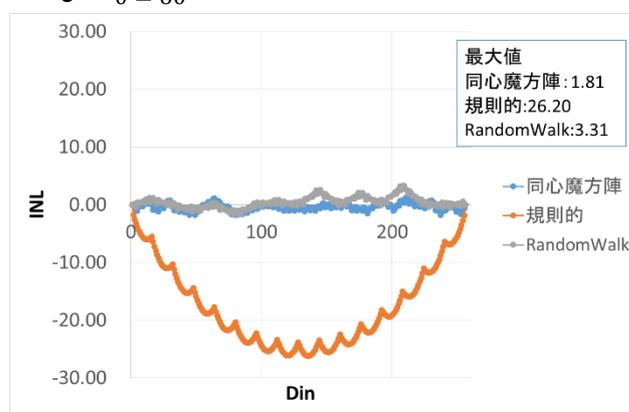


図 16 $\theta = 60^\circ$ の時のシミュレーション結果

Fig. 16. Simulated INL ($\theta = 60^\circ$)

一次のばらつきでは、Random Walk よりも同心魔方陣を用いたアルゴリズムの方が、ばらつきを軽減できたことが分かった。

2) 二次のばらつき

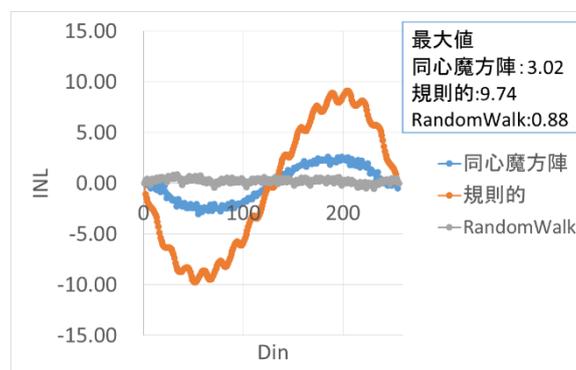


図 17 二次のばらつきのシミュレーション結果

Fig. 17. Simulated INL in case quadratic gradient error

二次のばらつきでは、同心魔方陣を用いるよりも Random Walk のアルゴリズムを用いたほうが、ばらつきを軽減できた。

3) 一次+二次のばらつき

1), 2) の結果より、二次よりも一次のほうが、ばらつきが大きい時、同心魔方陣のほうがばらつきを軽減できるのではないかと考えられた。また、二次の方が大きい場合は Random Walk のほうがばらつきを軽減できるのではないかと考えられたので、解析を行った。

- 一次 > 二次

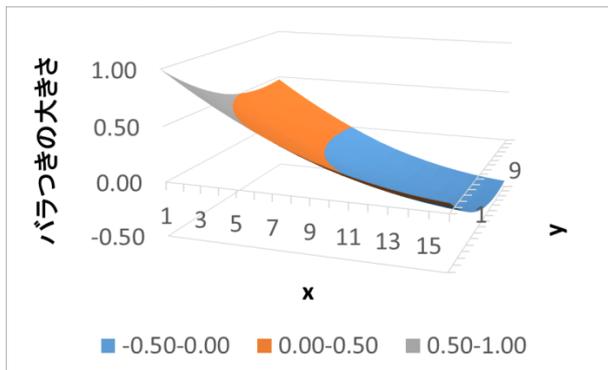


図 18 一次傾斜誤差が二次より大きい場合の分布
Fig. 18. Error distribution when linear gradient error is bigger than quadratic gradient error.

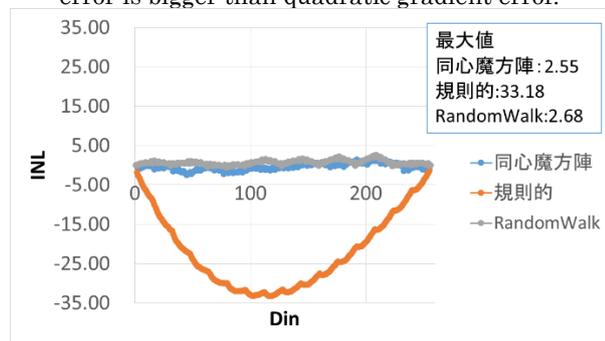


図 19 一次の方が二次より大きい場合の解析結果
Fig. 19. Simulated INL when linear gradient error is bigger than quadratic gradient error.

- 一次 < 二次

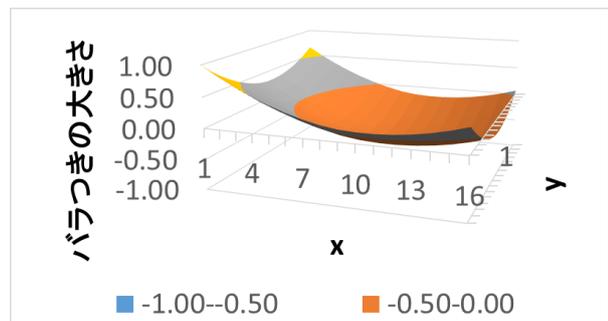


図 20 二次傾斜誤差が一次より大きい場合の分布
Fig. 20. Error distribution when quadratic gradient error is bigger than linear gradient error.

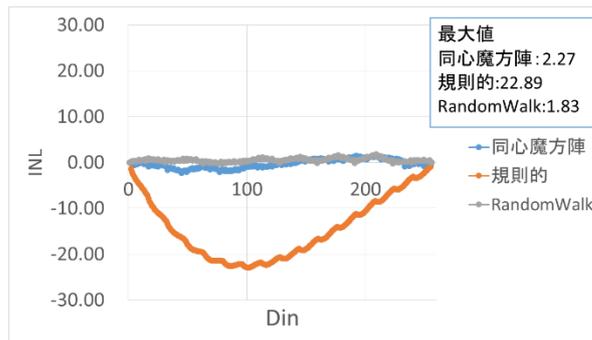


図 21 二次の方が一次より大きい場合の解析結果
Fig. 21. Simulated INL when quadratic gradient error is bigger than linear gradient error.

以上の結果より、一次の方が大きい場合は同心魔方陣のアルゴリズムが適しており、二次の方が大きい場合は Random Walk が適していると分かった。

同心魔方陣は、次にくる数字が対角線上に存在する傾向がある。このため、一次の場合は、正の値にばらついた電流源を選択した後、次の電流源は、対角線上にある負の値にばらついた電流源を選択するため、ばらつきをキャンセルすることができる。それに対して、二次の場合は対角線上の値は同じ符号のばらつきなために、一次と比べてばらつきをキャンセルできない。以上により、考案した同心魔方陣のアルゴリズムは、一次のばらつきに対して特化した結果となったのだと考えられる。

このため、一次+二次のばらつきの場合、一次の方が強い場合には同心魔方陣、二次の方が強い場合には Random Walk が適していたのだと考えられる。これより、一次と二次のばらつきをキャンセルするためには、ばらつきをキャンセルしつつ、様々な電流源を選択することにより、両者のばらつきを軽減できるのではないかと考えられる。

さらに魔方陣を用いて二次のばらつきを軽減する方法を考察した。同心魔方陣と Random Walk のアルゴリズムを比べると、Random Walk のほうが様々な点をスイッチングしていることから、同心魔方陣よりも擬似乱性の高い魔方陣を作ればよいのではないかと考えた。そこで対称魔方陣を用いたアルゴリズムを検討した。

6. 対称魔方陣による解析と考察

〈6・1〉対称魔方陣によるアルゴリズム

同心魔方陣よりも擬似乱性の高い魔方陣を再現するために、図 12 に示した対称魔方陣を 4 つ組み合わせると、図 22 の 8bit セグメント型 DA 変換器のレイアウトを考案した。B は A に対して右に 90°回転、C は A に対して右に 180°回転、D は A に対して右に 270°回転させたものである。

1. A の 1
2. B の 2
3. C の 3
4. D の 4

- 5. A の 5
- ⋮
- 63. C の 63
- 64. D の 64
- 65. B の 1
- 66. C の 2
- 67. D の 3
- 68. A の 4
- ⋮
- 255. B の 63
- 266. C の 64

以上のように各ブロックの電流源を右回りに順にスイッチングを行った。

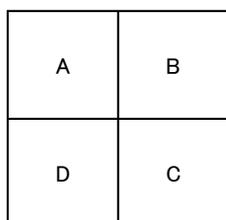


図 22 8bit 対称魔方陣を用いたアルゴリズム

Fig. 22. Algorithm using 8bit eighth-order symmetric magic square

〈6・2〉 解析結果と考察

1) 一次のばらつき

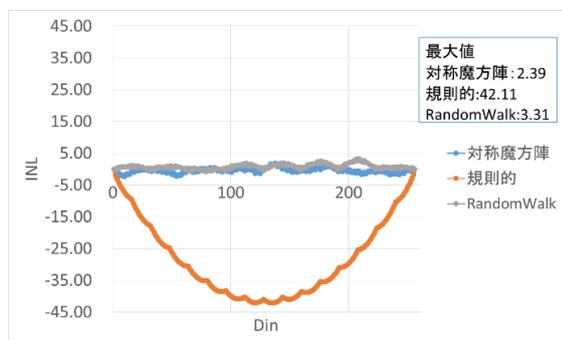


図 23 $\theta = 30^\circ$ の時のシミュレーション結果

Fig. 23. Simulated INL ($\theta = 30^\circ$)

2) 二次のばらつき

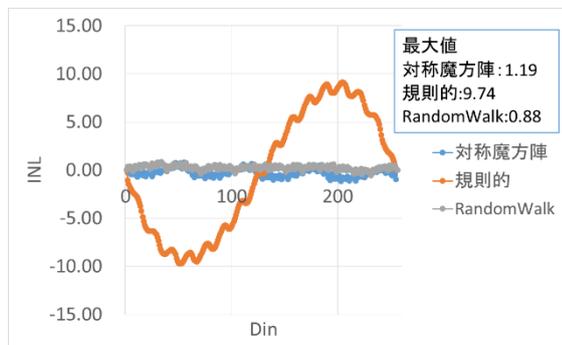


図 24 二次のばらつきのシミュレーション結果

Fig. 24. Simulated INL in case quadratic gradient error

擬似乱性を高めるために、4つの対称魔方陣を回転させて、全て異なった数字の配置にした。また、同心魔方陣に比べて対称魔方陣の方が、非規則に数字が配列されているため、二次のばらつきをキャンセルできるのではないかと考えた。

解析結果より、同心魔方陣の時と比べ、二次のばらつきを大幅に軽減することができた。しかし、同心魔方陣は対角線上に次の電流源を選択する特長があったことから、一次のばらつきに対しては、同心魔方陣のほうが特化していた。一次及び二次をキャンセルするには、対角線上に存在する電流源を選択し、かつ乱数性をもたせることが、システムティックなばらつきをキャンセルする方法なのではないかと考えられる。Random Walk よりもばらつきを軽減できていないため、さらに検討が必要である。

7. ばらつきの克服

一次及び二次のばらつきは、四隅に近づくほど大きくなる。これより、四隅を除いた電流源の配列にすることにより、線形性が向上される[4]。そこで、4次方陣を組み合わせ、図 25 のようなレイアウトを考案した。

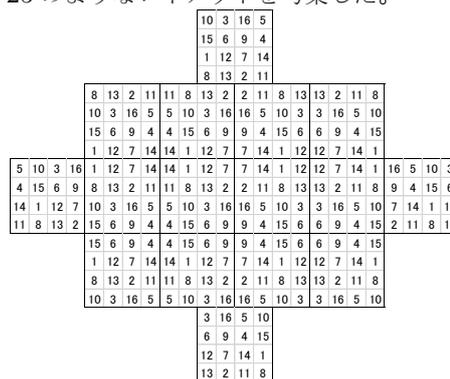


図 25 四隅を除いたレイアウト

Fig. 25. Layout which does not use 4 corner areas.

〈7・1〉 魔方陣によるアルゴリズム

図 26 に示した 4 次の魔方陣の配列より、以下のような順でスイッチングを行った。

- 1. A の 1
- 2. B の 1
- 3. C の 1
- ⋮
- 16. P の 1
- 17. A の 2
- ⋮
- 32. P の 2
- ⋮
- 256. P の 64

〈7・2〉 魔方陣と規則的にスイッチングした場合との比較
一次、二次のばらつきにおける INL を求めた。その時のシミュレーション結果を図 27, 28 に示した。

一次及び二次のばらつきは、四隅に行くほど大きくなる傾向がある。四隅を取り除いた電流源の配列にすることにより、四角形で電流源を配列していたときよりも、INL が大幅に改善された。

四隅を取り除くことにより、INL が改善されることが分かったが、今回 4 次方陣を 16 個用いて一つのセルと見立てて各々 1 つずつスイッチングしていくことにより、ランダムなスイッチングを実現したことも INL の改善になった一つの要因だと考えられる。また二次のシミュレーション結果において、Din が 120, 140 付近で INL が極度に上がっている領域がある。これは、セル上で比較的ばらつきが高いセルを選択してしまっているのが原因だと考えられる。このため、4 次方陣を回転や裏返しを行い、ばらつきが蓄積されないように分散させることを行えば、INL をより軽減できると考えられる。

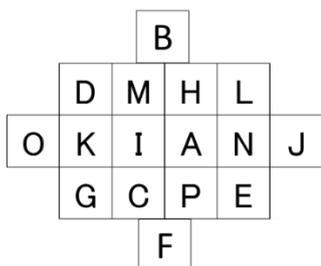


図 26 四隅を取り除いた電流源の配列

Fig. 26. Current cell array layout w/o 4 corner areas.

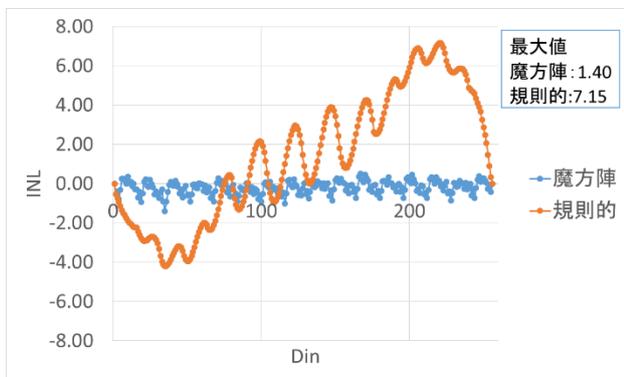


図 27 $\theta = 30^\circ$ の時の INL シミュレーション結果

Fig. 27. Simulated INL($\theta = 30^\circ$)

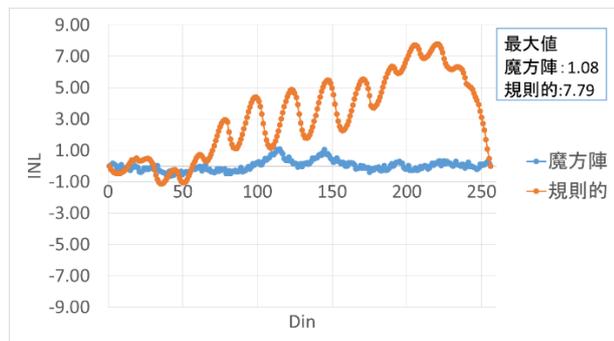


図 28 二次のばらつきの INL シミュレーション結果

Fig. 28. Simulated INL in case quadratic gradient error

8. サイの目方陣による解析と考察

7 節で四隅に近づくほどばらつきが大きくなり、線形性の劣化が進行してしまうことを示した。そこで、四隅を使用せずに DA 変換器のレイアウト方法が考案されている[3]。

魔方陣の一種である「サイの目方陣」を用いて四隅を使用しないセグメント型 DA 変換器を考案した。サイの目方陣とは、本来の魔方陣は数字でマス目を表しているが、数字の代わりに○印で示したものである。例えば数字が 3 であれば、○を 3 つで表すものである。サイの目方陣については、付録で詳しく説明する。

5 次のサイの目方陣を用いて 8bit の DA 変換器レイアウトを考案した。行・列変換を行った方陣を図 29 に示し、DA 変換器を図 30 に示した。○印は電流源を示している。空欄の箇所は、ダミーとする。5 次方陣をそのまま用いると電流源が 325 個になってしまい 8bit 以上になってしまうので、8bit になるようにマス目を調節した。256 個の○印があればよいので、69 個の○印を除くことになる。一番ばらつきの値が大きいのは四隅であるので、四隅のマスは使わないものとする。行・列の変換により 5, 17, 23, 24 のマスを四隅に移動させた。最もばらつきの少ない中心部には、25 を配置した。空欄の箇所はダミーとすることにより、ばらつきを分散させることができ、電流源を規則的に配列するよりも分散させた方が線形性の向上に繋がるのではないかと考えた。

魔方陣は、行・列・対角成分の定和性を満たしている条件があったが、用いた魔方陣は角線成分の定和性はないものとし、また○印の個数の定和性もないものとした。

17	24	1	8	15	➡	24	15	1	8	17
23	5	7	14	16		6	22	13	20	4
4	6	13	20	22		18	9	25	2	11
10	12	19	21	3		12	3	19	21	10
11	18	25	2	9		5	16	7	14	23

図 29 サイの目方陣の行・列変換

Fig. 29. Column-row transformation of dice magic square.

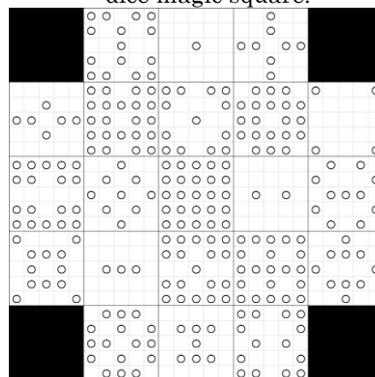


図 30 サイの目方陣による DA 変換器レイアウト

Fig. 30. DAC layout using dice magic square.

〈8・1〉サイの目方陣によるアルゴリズム

図 31 にスイッチングのアルゴリズムを示す。5 次のサイの目方陣であるので、25×25 のセルをもつ DA 変換器となる。25×25 のセルを 5×5 のブロックに分割する。5×5 の各ブロック A~U として考える（四隅にある 4 つのブロックはダミーである）。A ブロックの電流源を ON にした後、B ブロックの電流源を ON にする。A→B→C→...→U の順で各ブロックの電流源をデジタル信号に対応する分だけ一つずつ ON にしていく。ブロックのスイッチング順序は、1 から順に 25 まで ON にしていく。ただし、ダミーがある箇所は飛ばして次の電流源を ON にする。例えば、1 がダミーであれば、2 の電流源を ON にする。2 もダミーであれば 3 の電流源を ON にする。



図 31 サイの目方陣による DA 変換器レイアウト

Fig. 31. DAC layout using dice magic square.

〈8・2〉解析結果と考察

1) 一次のばらつき

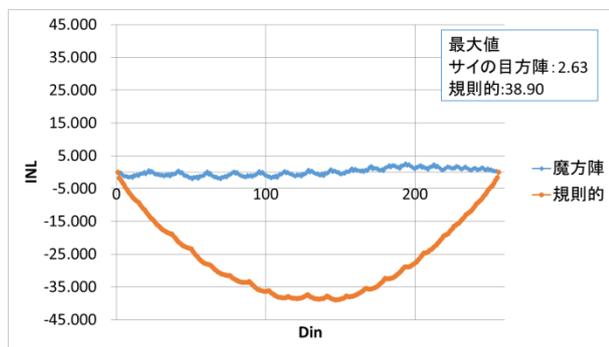


図 32 $\theta = 30^\circ$ の時の INL シミュレーション結果

Fig. 32. Simulated INL($\theta = 30^\circ$)

2) 二次のばらつき

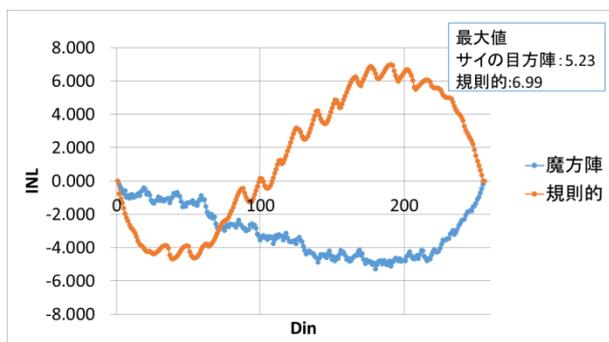


図 32 二次のばらつきの INL シミュレーション結果

Fig. 32. Simulated INL in case quadratic gradient error

最もばらつきの大きい四隅を取り除き、サイの目方陣によりスイッチを分散して疑似乱性を高めることにより、ばらつきの軽減を図った。一次のばらつきの INL は、同心魔方陣や対称魔方陣の結果とほぼ変わらなかった。二次の INL は同心魔方陣と対称魔方陣の 2 倍程度になった。サイの目方陣のマス目の数に対応するだけの電流源を配置し、デジタル入力に対応する分だけスイッチングを行うアルゴリズムとした。これにより、少数部分のセルと多数部分のセルが存在してしまっただけのため、スイッチングする箇所が多数部分に偏り、ばらつきを軽減できなかったのだと考えられる。

一次のばらつきは、図 31 より次に選択するセルを対称な位置に設置していたため、INL の値が左程変わらなかったのだと考えられる。しかし、後半になるにつれ選択するセルに偏りが生じてしまったため、二次のばらつきは軽減されなかったのだと考えられる。図 30 に示したレイアウトでは、空白の部分はダミーとしたが、ダミーの部分も利用することによりスイッチの配置の偏りがなくなり、 mismatches を軽減できるのではないかと考えている。二次のばらつきをキャンセルするには、魔方陣を用いて疑似乱を表現することが課題である。

9. CMOS イメージセンサへの応用

CMOS イメージセンサは、携帯電話やデジタルカメラに内蔵されている光を電気信号に変換するデバイスである。CCD イメージセンサが用いられていたが、近年低消費電力や小型化、CMOS プロセス技術が発展し、CMOS イメージセンサが用いられるようになった。カラーフィルターは、RGB の画素配列を周期的に組み合わせた 2×2 画素のベイヤー配列が用いられている。このカラーフィルターの前に光学ローパスフィルタを入れることによって、モアレや偽色の発生を防いでいる。しかし、ローパスフィルタにより高周波成分が劣化してしまう問題がある。現在の技術では、6×6 画素の非周期性を高めたカラーフィルターを用いることにより、モアレや偽色の発生を抑制し、光学ローパスフィルタを不要にして高周波成分の劣化を防いでいる。そこで、魔方陣を用いて非周期性を高めたカラーフィルターができるのではないかと考えている[9][10][11]。

10. まとめ

この論文では、魔方陣を用いて電流源を用いたセグメント型 DA 変換器の入出力関係の線形性向上を考案した。魔方陣の配列を工夫することにより疑似乱数を再現し、従来の技術である Random Walk よりも線形性が向上できる可能性があることを示した。今回、考案した魔方陣のアルゴリズムでは、魔方陣の次にくる数字がほぼ対称な位置にある特性によりばらつきをキャンセルすることが可能である特性を発見した。これにより、一次のばらつきに適していることを示した。

今後の展開として二次のばらつきにも対応した魔方陣を考えている。またデコーダ回路についても検討していく。さらに魔方陣を用いたレイアウト技術を、光学ローパスフィルタを用いない CMOS イメージセンサ[9][10][11]にも応用したいと考えている。

11. 謝辞

本研究をご支援いただいています半導体理工学研究センター(STARC)に感謝いたします。CMOS イメージセンサへの応用を示唆していただきました中谷隆之氏に感謝いたします。

文 献

- (1) 大森清美「魔方陣の世界」, 日本評論社 (2013年8月).
- (2) 佐藤肇「幾何学の魔術-魔方陣から現代数学」, 日本評論社 (2002年2月).
- (3) Xueqing LI, Qi WEI, Fei QIAO, Huazhong YANG, "Balanced Switching Schemes for Gradient-Error Compensation in Current-Steering DACs", IEICE Trans. Electron, vol.E95-C, no.11, pp.1790-1798 (Nov. 2012).
- (4) Yonghua Cong, Randall L. Geiger, "Switching Sequence Optimization for Gradient Error Compensation in Thermometer-Decoded DAC Arrays", IEEE Trans. Circuits and Systems II, vol.47, no.7 pp.585-595 (July 2000)
- (5) Geert A.M.Van der Plas, Jan Vandenbussche, Willy Sansen, Michel S.J.Steyaert, Georges G.E.Gielen, "A 14-bit Intrinsic Accuracy Q^2 Random Walk CMOS DAC", IEEE Journal of Solid-State Circuits, vol.34, no.12, pp.1708-1718 (Dec. 1999).
- (6) Ko-Chi Kuo, Chi-Wei Wu, "A Switching Sequence for Gradient Error Compensation in the DAC Design", IEEE Trans. Circuits and Systems II, vol.58, no.8 pp.502-506 (Aug 2011).
- (7) Takahiro Miki, Yasuyuki Nakamura, Masao Nakaya, Sotoju Asai, Youichi Akasaka, Yasutaka Horiba: "An 80-MHz 8-bit CMOS D/A Converter", IEEE Journal of Solid-State Circuits, vol.21, no.6 pp.983-988 (Dec. 1986).
- (8) 谷口研二、「CMOS アナログ入門」, CQ 出版社 (2011年6月)
- (9) 「トランジスタ技術 SPECIAL カメラ・モジュールの動かし方と応用製作」, CQ 出版社 (2013年10月)
- (10) 江尻正員, 奥富正敏他 (著): 「デジタル画像処理」, CG-ARTS 協会 (2011年3月)
- (11) 「富士フイルム X-Trans CMOS センサー」
<http://fujifilm-x.com/x-pro1/ja/about/sensor/> (2014/08/28 アクセス)

付録: 様々な魔方陣

魔方陣は、今回取り上げた同心魔方陣や対称魔方陣以外にも様々な種類が存在する。それらの魔方陣には、個性があり、今後の研究で応用が出来ると考えられるので、付録に示す[1][2]。

1) フランクリンの魔方陣

8次のフランクリンの魔方陣を図32に示す。この8次魔方陣は、対角線上において定和性を満たしておらず、魔方陣としては不完全であるが、興味深い性質を持っている。

52	61	4	13	20	29	36	45
14	3	62	51	46	35	30	19
53	60	5	12	21	28	37	44
11	6	59	54	43	38	27	22
55	58	7	10	23	26	39	42
9	8	57	56	41	40	25	24
50	63	2	15	18	31	34	47
16	1	64	49	48	33	32	17

図32 フランクリンの魔方陣

Fig. 32. Franklin's magic square

- 各半行・列(4個の数から成る)の数の和は、全て130になる。したがって、4分割した4つの4次配列、6次配列の全要素の総和も一定である。
- 任意の2次配列の4数の和は130である。このような2次配列は49組存在する。したがって、任意の4次配列、6次配列の全要素の総和も一定である。これを、相結型の魔方陣という。
- 図33において、左下隅の16から斜めに10まで上がり、次に右隣の23から右下隅の17まで下がると、その8数の和は定和260に一致する。これと平行な全ての折れ線上の8数の和も成立する。また、この反転した形も定和が成立する。この性質をフランクリン型と呼ぶ。

52							45
	3					30	
		5			28		
			54	43			
			10	23			
			57			40	
	63						34
16							17

図33 フランクリン型定和性

Fig. 33. Franklin constant sum characteristics.

- 図34に示す図形に含まれる8数の和は260である。8次方陣のどこを取っても、この8組が定和260を与える。

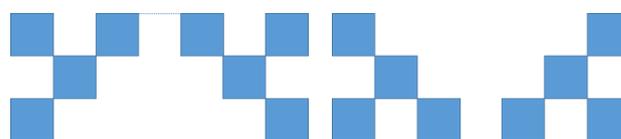


図34 8組の定和性

Fig. 34. Constant sum of combination of 8 numbers.

- 図 35 に示す図形に含まれる 8 数の和は 260 である。四隅の 4 数と中央の 4 数の和は 260 である。

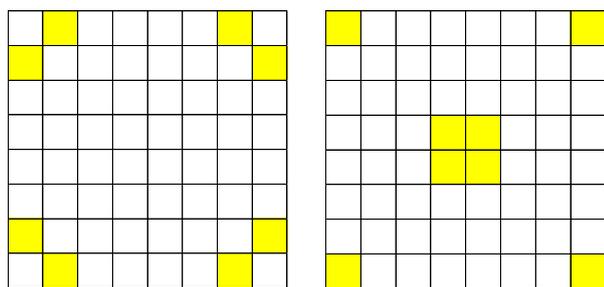


図 35 偶の定和性及び、4 隅と中央の定和性

Fig. 35. Franklin's magic square constant sum characteristics in 4 corners and center.

一つの魔方陣に、これほど多くの性質を持ったものは珍しい。なお、フランクリンの魔方陣は 16 次方陣も存在する。

2) サイの目魔方陣

サイの目方陣とは、魔方陣の数の代わりに○印で表したものである。図 36 に 3 次のサイの目方陣を示した。図を 9×9 配列と考えると、各行・列・対角成分の○印の個数は等しい。3 次の場合は、○印の個数は 5 個となっている。また、サイの目方陣を作る際、各数を表す○印の図形は、中心に関して左右上下に対称性をもつものとする。

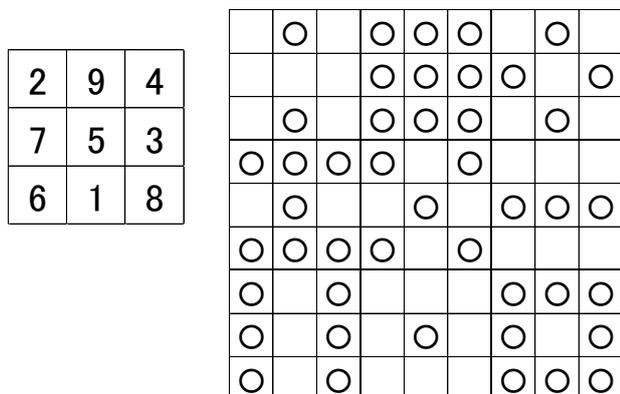


図 36 サイの目方陣の例

Fig. 36. Example of dice magic square.

4) 素数方陣

素数だけを用いて作られた魔方陣を素数方陣と呼ぶ。素数とは、異なる 2 つの約数をもつ数字である。ゆえに、最小の素数は 2 である。しかし、2 を含む素数方陣は存在しない。素数の中で 2 だけが偶数であり、他は奇数であるためである。これにより、2 を含む行・列と含まない行・列の要素の和は、偶数と奇数に別れてしまうためである。よって、素数方陣で用いられる最小の素数は 3 である。

3 で始まる連続した素数を用いて素数方陣を作ると、最小の次数は 35 次である。35 次素数方陣に含まれる最大数は、9941、定和は 163043 である。35 次素数方陣は、膨大な大きさになってしまうため、省略する。

素数方陣の条件を緩くし、1 も素数として扱うことによ

り、魔方陣の規模を縮小できる。1 を用いたときの最小次数は、12 次である。12 次素数方陣を図 37 に示す。総和は 54168、定和は 4514 である。

さらに、最小素数や連続素数を用いる条件をはずすことにより、最小定和やいろいろな性質（対称方陣、同心方陣など）を満たす魔方陣を作ることが可能となる。図 38 に定和性を満たした魔方陣、図 39 に対称魔方陣、図 40 に同心魔方陣を示した。

367	557	449	631	823	571	11	53	347	167	179	359
773	373	283	127	641	739	827	5	193	113	163	277
293	281	379	461	563	643	457	751	7	61	419	199
211	227	311	383	569	467	647	743	757	3	59	137
223	17	229	307	389	463	727	653	661	761	13	71
23	233	139	331	157	397	479	677	577	659	769	73
19	151	79	673	239	317	401	487	599	619	197	733
149	787	97	83	67	491	313	409	241	593	601	683
691	29	797	89	101	173	337	251	131	499	809	607
613	701	509	31	103	37	271	257	547	421	521	503
709	617	523	811	41	107	43	181	263	349	431	439
443	541	719	587	821	109	1	47	191	269	353	433

図 37 12 次の素数方陣

Fig. 37. Twelfth-order prime numbers magic square.

31	73	7
13	37	61
67	1	43

23	71	5	3
31	13	41	17
1	11	37	53
47	7	19	29

図 38 素数方陣

Fig. 38. prime numbers magic square.

13	97	83	47
41	89	103	7
113	17	31	79
73	37	23	107

図 39 対称素数方陣

Fig. 38. Symmetric prime numbers magic square.

401	11	263	149	431
419	461	23	269	83
311	59	251	443	191
53	233	479	41	449
71	491	239	353	101

図 38 同心素数方陣

Fig. 38. Concentric prime numbers magic square