DAC LINEARITY IMPROVEMENT WITH LAYOUT TECHNIQUE USING MAGIC AND LATIN SQUARES

Dan Yao, Yifei Sun, Masashi Higashino, Shaiful Nizam Mohyar*[2]
Tomonori Yanagida, Takuya Arafune, Nobukazu Tsukiji, Haruo Kobayashi

Division of Electronics and Informatics, Gunma University, Kiryu 376-8515 Japan

[2] School of Microelectronic Eng., University Malaysia Perlis, Pauh Putra Campus, Arau, Perlis, 02600 Malaysia

* t171d087@gunma-u.ac.jp

ABSTRACT

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

Index Terms— *DAC, Magic Square, Latin Square, Linearity, Layout, Mismatch*

1. INTRODUCTION

Because of the demand for small-sized high-speed electronic devices, digital circuits should be adopted to the requirements. Along with the progress of digitization, many electronic devices are equipped with digital-to-analog converters (DACs) [1, 2]. Semiconductor devices on silicon wafer suffer from random and systematic mismatches regarding to the characteristics of MOSFETs, resistors, and capacitors [3], which can make the input and output relationships of the DAC non-linear.

In this paper we present layout algorithms based on magic square and Latin square [4-9] properties to cancel the systematic mismatches among unit cells and improve the DAC overall linearity. Magic and Latin squares have been investigated from long time ago by many mathematicians, and there are a lot of theoretical research results. However, their applications of analog/mixed-signal IC design have not been reported except for from our group [14], to our knowledge. The proposed layout methods are expected to be refined further by using the theoretical results for magic and Latin squares.

We remark that in our previous publication [14], the magic square algorithm is applied to sorting of the unit cells to improve the unary DAC linearity. However, this paper here describes unit cell layout algorithms based on magic and Latin squares. Hence, these are very different.

2. CONFIGURATION AND OPERATION OF SEGMENT DAC

DAC architectures may be classified into binary and unary types as well as their combination (segment type), as shown in Fig. 1 [1, 2]. The binary one uses the sum of the binary element outputs as its overall output while the unary sums the unary outputs obtained from the decoded binary data. The unary DAC consists of the small units of voltage, charge or current. DA conversion is performed by summing these unit cell outputs. Fig. 2 shows a unary DAC with unit current sources, which turn on according to the corresponding decoded digital data. Then, the digital input is converted into an analog output.

The unary type has less influence on the output signal than the binary type even if there are mismatches among elements. The glitch is small and monotonicity characteristics can be guaranteed in principle. However, its disadvantages are large hardware and power due to a large number of unit cells, as well as conversion speed restriction to the decoder circuit. When attempting to realize a high linearity DAC, the relative mismatches among the unit cells (unit currents I in Fig. 2) become a problem. Thus, the layout technique for alleviating this influence is necessary.

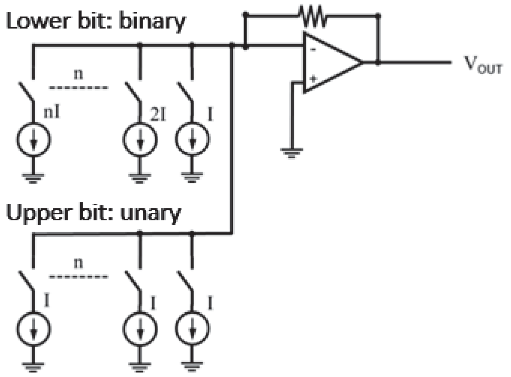
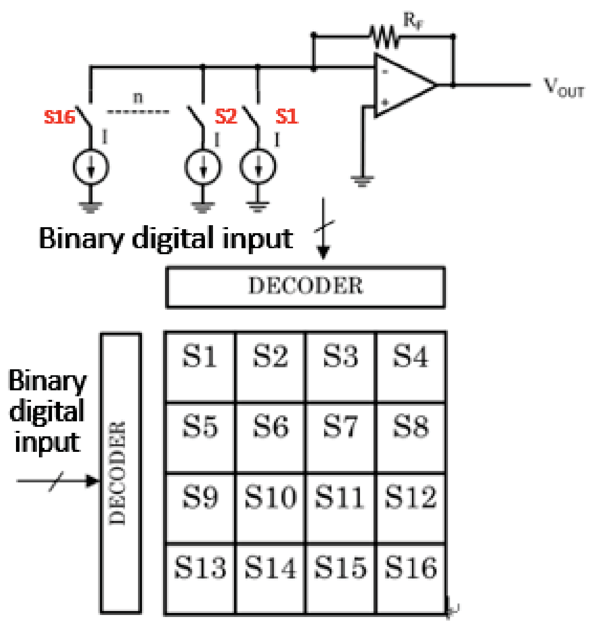


Fig. 1. Segmented DAC

978-1-5386-2159-2/17/$31.00 ©2017 IEEE



S1	S2	S3	S4
S5	S6	S7	S8
S9	S10	S11	S12
S13	S14	S15	S16

Fig. 2. Unary DAC circuit and layout of unit cell array.

Many DACs are combination of unary and binary types. Unary type with low sensitivity devices is used for the higher bits, while the binary type with a small number of elements for lower bits. Then a high performance DAC with appropriate circuit scale and power can be realized.

Then we consider only the unary type because the unary type handles higher bits, which influences overall linearity of the segmented DAC.

3. VARIATIONS IN CIRCUIT ELEMENT CHARACTERISTICS

There are systematic variations among MOSFET, resistor and capacitor characteristics on an integrated circuit, due to their placements (locations), and random variations which do not depend on their placements. Ideally, the input and output characteristics of the DAC should be linear. However, in reality due to these variation effects, it can be nonlinear. The causes of these variations are as follows [2, 9-13].

1) Systematic variations
- Voltage drop on wiring
- Temperature distribution
- CMOS manufacturing process
- Doping distribution
- Changes in threshold voltage
  - Accuracy in wafer plane
  - Mechanical stress

2) Random variation
- Device mismatch.

The systematic variations can be modeled as linear and quadratic gradients as well their combination, regarding to the circuit element placements.

1) Linear gradient variation
- Voltage drop on wiring
- CMOS manufacturing process

2) Quadratic gradient variation
- Temperature distribution
- Accuracy in wafer plane
- Mechanical stress.

The above variations largely affect the DAC linearity. The variations among the unit current sources are shown in Figs. 3, 4 and 5. (x, y) is assumed to be the coordinates of the position on the chip, and the linear and quadratic variations are shown by the following expressions.

1) Linear error (Fig. 3)

$$\varepsilon_l(x,y) = g_l * \cos\theta * x + g_l * \sin\theta * y$$

θ: Angle of inclination, g_l: Magnitude of the slope

2) Quadratic error (Fig. 4)

$$\varepsilon_q(x,y) = g_q * (x^2 + y^2) - a_0$$

g_q: Variable quantity, a_0: Position

3) Linear and quadratic joint errors (Fig. 5)

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

The influence of the systematic variation on the DAC linearity can be mitigated by the layout technique for the unit cells. In the case of the segmented DAC, the variation effects may be reduced by the conventional method (random walk). Then it also improves the linearity.

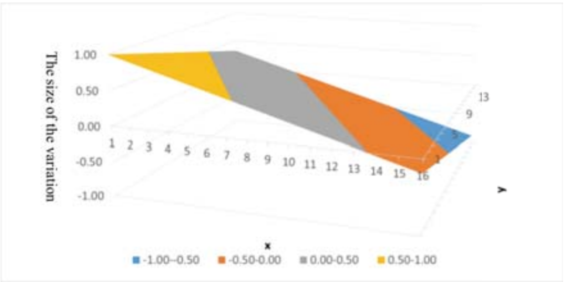


Fig. 3. Linear gradient error

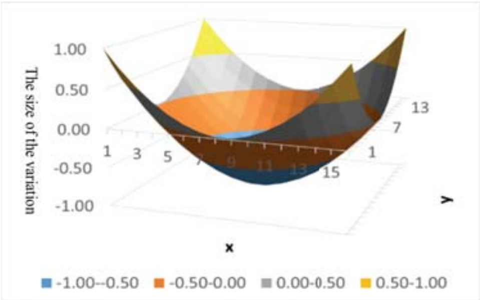


Fig. 4. Quadratic gradient error

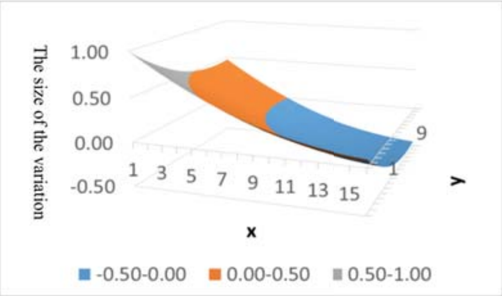


Fig. 5. Linear and quadratic joint gradient errors

4. UNARY DAC NONLINEARITY

In practical CMOS technologies, the current source mismatches are influenced by their threshold voltage mismatch and/or by the slope mismatch (Fig.5) [3]. Ideal drain current I_d in saturation region is given by:

$$I_{d,1} = \frac{}{2}\left(V_{gs} \quad V_{th1}\right)^2$$

Also drain current mismatch is given by :

$$\frac{I_{d1}}{I_{d1}} = \frac{2}{V_{gs} \quad V_{th1}} \frac{Av_{th}t_{ox}}{\sqrt{WL}}$$

Current mismatches are dependent on their device sizes. Note that here, we are considering to reduce the DAC nonlinearity effects of the current mismatches due to small device size $\left(\sqrt{WL}\right)$.

These mismatches among current sources may cause the unary DAC nonlinearity due to their systematic mismatches if they are laid out in a regular manner (Figs. 6, 7).

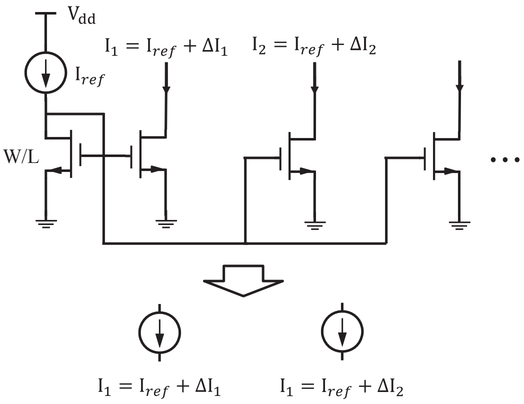
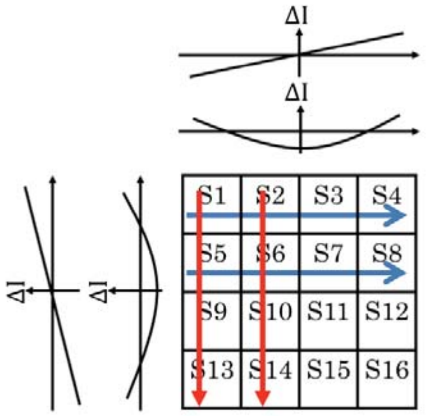
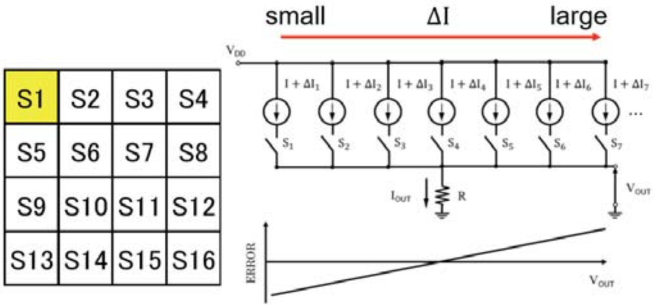

V_{dd}
$I_1 = I_{ref} + \Delta I_1$
$I_2 = I_{ref} + \Delta I_2$
I_{ref}
W/L
…
$I_1 = I_{ref} + \Delta I_1$
$I_1 = I_{ref} + \Delta I_2$

Fig. 5 Mismatches among current sources.



S1	S2	S3	S4
S5	S6	S7	S8
S9	S10	S11	S12
S13	S14	S15	S16

Fig. 6 Regular layout of unit cells for a unary DAC and their linear/quadratic gradient errors

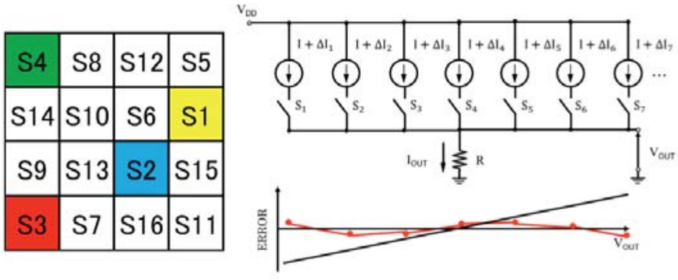


S1	S2	S3	S4
S5	S6	S7	S8
S9	S10	S11	S12
S13	S14	S15	S16

Fig. 7 Regular layout of unit cells for a unary DAC and its nonlinearity due to their linear gradient errors

4. MAGIC SQUARE LAYOUT TECHNIQUE

A magic square has a property that the sums of each row/ column/diagonal elements are all equal. Hence we consider that this property balances the unit cell array of the unary type DAC, and we have investigated the layout of the unit cells to reduce the systematic variation effects to improve the DAC linearity (Fig. 8).



S4	S8	S12	S5
S14	S10	S6	S1
S9	S13	S2	S15
S3	S7	S16	S11

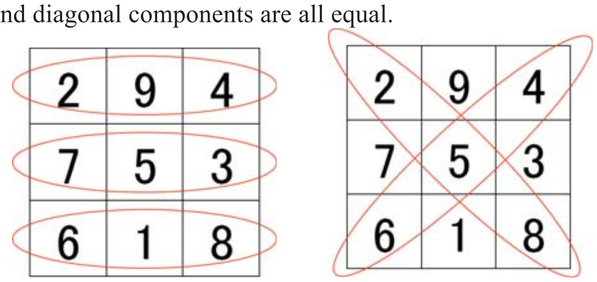
Fig. 8 Layout technique of unit cells for a unary DAC and its linearity improvement by cancelling their linear gradient errors

4.1. Features of Magic Square

The magic square with an *n* x *n* matrix is a series of natural numbers starting from 1 to *n* x n, arranged in a grid pattern, and the numbers on each row, column or diagonal elements have equal sum [4-6]. Including n elements on each row, column, diagonal, the magic square is usually called as an n-th order magic square. The sum of the rows, columns and diagonal elements of the n-th order magic square is expressed as follows:

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

In the magic square shown in Fig. 9, it can be confirmed that the sum of the elements of each row, column and diagonal components are all equal.



2	9	4
7	5	3
6	1	8

2	9	4
7	5	3
6	1	8

Fig. 9. Equivalent constant sum characteristics.

Utilizing this property, the systematic variations are expected to be reduced by the magic square layout of the unit cells for a segmented DAC. Even if a concentric magic square is removed from the outside of the magic square by one side, the remaining part always does not lose its integrity. The 8-th concentric magic square used in the analysis is shown in Fig. 10. It is realized by combining four squares with an 8-bit segmented DAC.

4.2. Algorithm Using Concentric Magic Square

An 8-bit square combining of four 6-bit concentric magic squares which have good symmetry is shown in Fig. 10, and the DAC linearity improvement with the layout based on the magic square was confirmed by simulation.

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

Fig. 10. Four 6-bit concentric magic squares

4.3. Analysis Results and Consideration

4.3.1. Linear gradient error

We have examined the linear gradient error case for the unary DAC and its INL (integral nonlinearity) was examined by simulation for several layout algorithms. We see that in the linear variation, the concentric magic square was effective to reduce the variation effects (Fig. 11).

4.3.2. Quadratic gradient error

In the quadratic gradient error case, the magic square algorithm is effective, compared to the conventional regular layout algorithm (Fig. 16), and also another layout algorithm (random walk algorithm [11, 12]) is also effective. (Fig. 12).

4.3.3 Linear and Quadratic joint gradient errors

When the linear gradient variation is larger, the concentric magic square algorithm will be effective (Fig. 13), and when the quadratic is larger, the random walk is effective (Fig. 14).

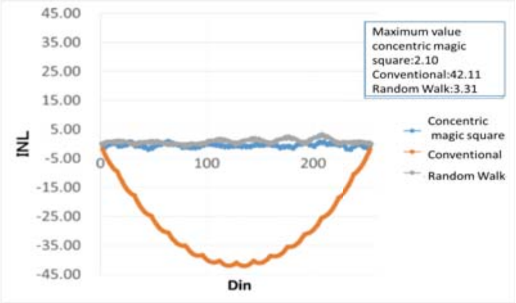


Fig. 11. Simulated INL in the linear gradient error case.

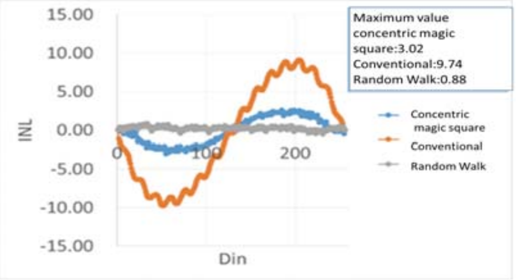


Fig. 12. INL in the quadratic gradient error case

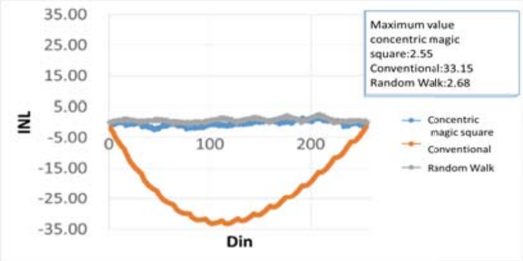


Fig. 13. Simulated INL when the linear gradient error is bigger than the quadratic gradient error.

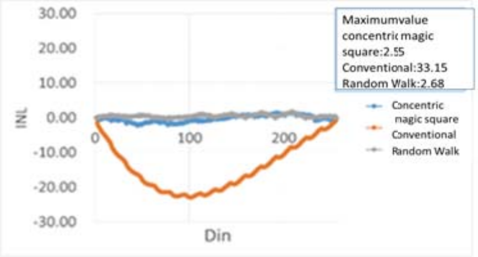


Fig. 14. Simulated INL when the quadratic gradient error is bigger than the linear gradient error.

5. LATIN SQUARE LAYOUT ALGORITHM

Latin squares have n rows and n columns of n different symbols arranged in such a way that each symbol appears only once in each row and each column (Fig. 15 (a)) [6-8]. Now considering a “complete Latin square”; for even n，put the numbers 1 through n in the first row in the following order：1, 2, n, 3, n-1,…., n/2+2, n/2+1. (For example, for n=4, we would have 1, 2, 4, 3, …). In each remaining empty cell, we place the number in the cell directly above it plus 1 (Fig. 15 (b)).

The Latin square was studied by mathematician Leonhard Euler (1707-1783).

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

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

Fig.15 (a) 4x4 Latin square (b) 4x4 complete Latin square

The linearity of the unary DAC in the variations of the linear and quadratic gradients, is degraded in the regular layout (Fig. 16) in case of 8-bit (16 × 16). This regular layout, the common centroid layout (Fig. 17) [10, 11, 12] and the complete Latin square layout (Fig. 18) are compared.

We have simulated static and dynamic performances of an 8-bit unary DAC, and comparison among the regular layout, the common centroid layout and the complete Latin square layout was performed. The static performance was simulated as INL and DNL (Figs. 19-20). The simulated dynamic performance was obtained as Spurious Free Dynamic Range (SFDR) [1, 2] (Figs. 21-23); there mismatch of current sources was generated as a random number between -1 from +1. We see that the DAC with the Latin square algorithm improved SFDR.

The numerical simulation results in the linear gradient variation case are shown in Fig. 19 and the results in the quadratic gradient variation are shown in Fig. 20. We see that the Latin square and the common centroid methods make the unary DAC to have almost equal linearity.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176
177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208
209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256

Fig. 16 Regular layout of 2D array of current cells

1	3	6	7	15	9	12	13	4	5	8	2	10	11	14	16
3	2	4	5	8	16	10	11	6	7	1	9	12	13	15	14
6	4	2	3	5	7	16	9	8	1	10	12	14	15	13	11
7	5	3	1	4	6	8	15	2	9	11	13	16	14	12	10
15	8	5	4	1	3	6	7	10	11	14	16	13	12	9	2
9	16	7	6	3	2	4	5	12	13	15	14	11	10	1	8
12	10	16	8	6	4	2	3	14	15	13	11	9	1	7	5
13	11	9	15	7	5	3	1	16	14	12	10	2	8	6	4
4	6	8	2	10	12	14	16	1	3	5	7	15	9	11	13
5	7	1	9	11	13	15	14	3	2	4	6	8	16	10	12
8	1	10	11	14	15	13	12	5	4	2	3	6	7	16	9
2	9	12	13	16	14	11	10	7	6	3	1	4	5	8	15
10	12	14	16	13	11	9	2	15	8	6	4	1	3	5	7
11	13	15	14	12	10	1	8	9	16	7	5	3	2	4	6
14	15	13	12	9	1	7	6	11	10	16	8	5	4	2	3
16	14	11	10	2	8	5	4	13	12	9	15	7	6	3	1

Fig. 17 Common centroid layout of 2D array of current cells

1	2	N	3	N-1	4	n-2	5	n-3	6	n-4	7	n-5	8	n-6	9

1	2	16	3	15	4	14	5	13	6	12	7	11	8	10	9
2	3	1	4	16	5	15	6	14	7	13	8	12	9	11	10
3	4	2	5	1	6	16	7	15	8	14	9	13	10	12	11
4	5	3	6	2	7	1	8	16	9	15	10	14	11	13	12
5	6	4	7	3	8	2	9	1	10	16	11	15	12	14	13
6	7	5	8	4	9	3	10	2	11	1	12	16	13	15	14
7	8	6	9	5	10	4	11	3	12	2	13	1	14	16	15
8	9	7	10	6	11	5	12	4	13	3	14	2	15	1	16
9	10	8	11	7	12	6	13	5	14	4	15	3	16	2	1
10	11	9	12	8	13	7	14	6	15	5	16	4	1	3	2
11	12	10	13	9	14	8	15	7	16	6	1	5	2	4	3
12	13	11	14	10	15	9	16	8	1	7	2	6	3	5	4
13	14	12	15	11	16	10	1	9	2	8	3	7	4	6	5
14	15	13	16	12	1	11	2	10	3	9	4	8	5	7	6
15	16	14	1	13	2	12	3	11	4	10	5	9	6	8	7
16	1	15	2	14	3	13	4	12	5	11	6	10	7	9	8

Fig. 18 Complete Latin square layout of 2D array of current cells

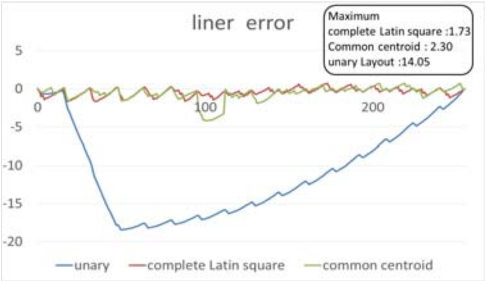


Fig. 19 INL in the linear gradient error case.

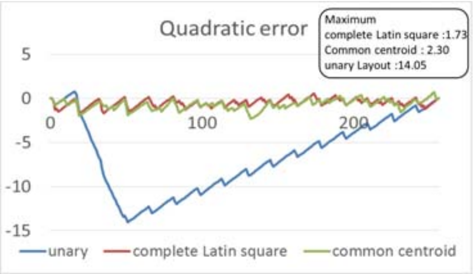


Fig. 20 DNL in the linear gradient error case.

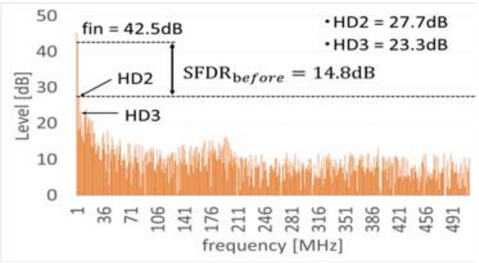


Fig. 21 SFDR in the regular layout case.

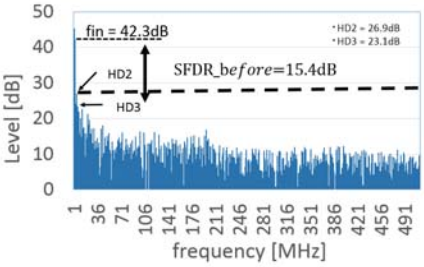


Fig. 22 SFDR in the common centroid layout case.

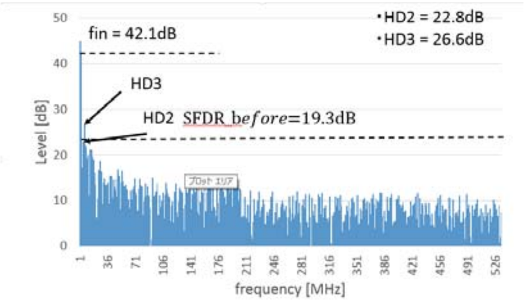


Fig. 23 SFDR in the complete Latin square case.

6. CONCLUSION

In this research, we devise an improvement that magic square and Latin square layout algorithms are used for the layout algorithms among unit cells to improve the linearity of a segmented DAC by suppressing their systematic mismatch effects. Pseudo random numbers are reproduced by the arrangement of the magic square and Latin square. In addition, it is indicated that the linearity can be improved more than the conventional technique.

We expect that since there are a lot of magic squares and Latin squares with useful properties and their mathematical research results, the layout algorithms using them will be refined further.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Takahiro Miki for valuable suggestions.

REFERENCES

[1] F. Maloberti, Data Converters, Spring (2007),

[2] R. V. D. Plassche, Integrated Analog-to-Digital and Digital-to-Analog Converters, Springer (2012).

[3] M. J. M. Pelgrom, AAD C. J. Duinmaijer, A. O. G. Welbers, “Matching Properties of MOS Transistors”, IEEE Journal of Solid-State Circuits, vol. 24, no.5 , pp.1433-1440 (Oct. 1989).

[4] K. Omori, The World of Magic Square, Nippon Hyoron Sha, Japan (Aug 2013).

[5] H. Sato, Geometry Magic-Modern Mathematics From the Magic Square, Nippon Hyoron Sha, Japan (2002).

[6] T. Omura, Story of Mathematical Puzzle, Union of Japanese Scientists and Engineers (1998).

[7] M. Yoshizawa, How to Take a Problem, How to Use Practical Application, Maruzen (2012).

[8] W. L. Stevens, “The Completely Orthogonalized Latin Square”, vol. 9, no. 1, 82-93 (Jan 1939).

[9] T. Miki, Y. Nakamura, M. Nakaya, S. Asai, Y. Akasaka, Y. Horiba., "An 80-MHz 8-bit CMOS D/A Converter”, IEEE Journal of Solid-State Circuits, vol. 21, no. 6, pp.983- (Dec. 1986).

[10] G. A. M. Van der Plas, J. Vandenbussche, W. Sansen, M. S. J. Steryaert, and G. G. E. Gielen, “A 14-bit Intrinsic Accuracy Q2 Random Walk CMOS DAC”, IEEE Journal of Solid-State Circuits, vol. 34, no. 12, pp. 1708-1718 (Dec. 1999).

[11] Y. Cong, R. L. Geiger, “Switching Sequence Optimization for Gradient Error Compensation in Thermometer-Decoded DAC Arrays”, IEEE Trans. Circuits and Systems II, vo. 47, no. 7, pp.585- 595 (July 2000)

[12] K.-C. Kuo, C. –W. 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).

[13] X. Li, F. Qiao, H. 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).

[14] M. Higashino, S. N. B. Mohyar, H. Kobayashi, “DAC Linearity Improvement Algorithm With Unit Cell Sorting Based on Magic Square”, IEEE International Symposium on VLSI Design, Automation and Test, Hsinchu, Taiwan (April, 2016).