Journal of Mechanical and Electrical Intelligent System (JMEIS)

Segmented DAC Unit Cell Selection Algorithm and Layout/Routing Based on Classical Mathematics

Dan Yao[1,a,*], Xueyan Bai[1,b], Anna Kuwana[1,c], Kazuyuki Kawauchi[1,d], Yujie Zhao[1,e], Jianglin Wei[2,f], Shogo Katayama[1,g], Masashi Higashino[1,h], Haruo Kobayashi[1,i]

[1]Division of Electronics and Informatics, Faculty of Science and Technology, Gunma University
1-5-1 Tenjin-cho, Kiryu, Gunma, 376-851, Japan

[2]Faculty of Intelligence Manufacturing, Yibin University, Yibin Sichuan, 64400 China

*Corresponding author

[a]<t202d005@gunma-u.ac.jp>, [b]<t202d004@gunma-u.ac.jp> , [c]<kuwana.anna@gunma-u.ac.jp>, [d]<kawauchi.kazuyuki@gmail.com>, [e]<t202d002@gunma-u.ac.jp>, [f]<t171d601@gunma-u.ac.jp>, [g]<t202d601@gunma-u.ac.jp>, [h]<t15804080@gunma-u.ac.jp>, [i]<koba@gunma-u.ac.jp>

Keywords: DAC, systematic mismatch, linearity, Euler Knight Tour, Knight Tour, Magic Square, Latin Square

Abstract. Semiconductor devices on silicon wafers suffer from random and systematic mismatches regarding the characteristics of MOSFETs, resistors, and capacitors, which can make the input and output relationship of the DAC non-linear. This paper investigates unit cell selection algorithms based on the Euler Knight tour algorithm as one of the random selection algorithms. The Euler Knight tour has properties of both the Knight tour and the Magic square. We show that it can improve the linearity of a segmented digital-to-analog converter (DAC) by cancelling systematic mismatch effects among unit cells, and we also show its comparison with other methods (Magic square and Latin square algorithms). Further their layout and routing feasibilities are investigated.

1. Introduction

Because of the demand for small-sized high-speed electronic devices, digital-oriented electronic circuits/systems should be adopted to meet the requirements in the IoT era. Along with the progress of digitization, many electronic devices are being equipped with digital-to-analog converters (DACs) because real-world signals are analog [1, 2]. However, semiconductor devices on silicon wafers suffer from random and systematic mismatches regarding the characteristics of MOSFETs, resistors, and capacitors [3], which can make the input and output relationship of the DAC non-linear.

In this paper we investigate a unit cell selection algorithm for a segmented DAC, based on the Euler Knight tour, which has the benefits of both the Magic square and the Knight tour [4-9]; it cancels the systematic mismatches among unit cells and improves overall DAC linearity. The Latin square, Magic square and Knight tour algorithms have been investigated for a long time ago by many mathematicians, and there are a lot of theoretical research results. However, there are very few reports on their application to analog/mixed-signal IC design except from our group [14-16] and a Russian group [17]. The proposed algorithm is expected to refine the DAC linearity improvement algorithm further according to the theoretical results for Magic square and Knight tour.

The Knight tour algorithm is expected to select the next unit cell close to the present cell so that the differential non-linearity (DNL) would not be large; if the next selected cell is far from the present cell, the DNL can be large due to systematic mismatch. The Magic square algorithm, on the other hand, offers well-balanced cell selection [18]. The Knight tour algorithm is good for local placement while

the Magic square algorithm is for global placement, and hence the Euler Knight tour algorithm is expected to be good in terms of both local and global placement.

We note that our previous work [14-16] applied the Magic square algorithm to unit cell sorting to improve unary DAC linearity. The Russian group at Saint Petersburg investigated the Knight tour algorithm [17], but not Euler Knight tour algorithm, even though Leonhard Euler stayed at Saint Petersburg for a long time. This paper describes unit cell selection algorithms based on the Euler Knight tour which has both properties of Knight tour and Magic square; they look similar but are different.

We have also developed a layout and routing design EDA tool for 2D unit cell arrays for regular, Magic square, Latin square and Euler Knight tour algorithms, and show here their feasibilities for DAC implementation on an IC. Both advances are necessary because Magic square and Euler Knight tour algorithms make layout and routing more complicated than the regular unit cell selection algorithm.

The first technique for systematic mismatch effect reduction that used the unit cell selection algorithm for unary DAC linearity improvement was proposed in [9]. Subsequently, there have been several studies in this area [10-17]. Our proposal targets better 2D pseudo randomization, though layout and routing are complicated compared to [9].

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 type sums the binary element outputs to create its overall output while the unary sums the unary outputs obtained from the decoded binary data. The unary DAC consists of small units of voltage, charge or current, and 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 in thermometer code format. This converts the digital input into an analog output.

In terms of DNL influence on the output signal, the unary type is more robust than the binary type even if there are mismatches among elements. Any glitch is small and monotonicity characteristics can be guaranteed in principle. However, its disadvantages are relatively large hardware and power consumption due to a large number of unit cells, as well as conversion speed restrictions due to the decoder circuit. When attempting to realize a high linearity DAC, the relative mismatches among the unit cells (unit currents denoted as *I* in Fig. 2) become a problem; they cause overall DAC nonlinearity. Thus, a cell selection technique that can alleviate this influence is necessary.

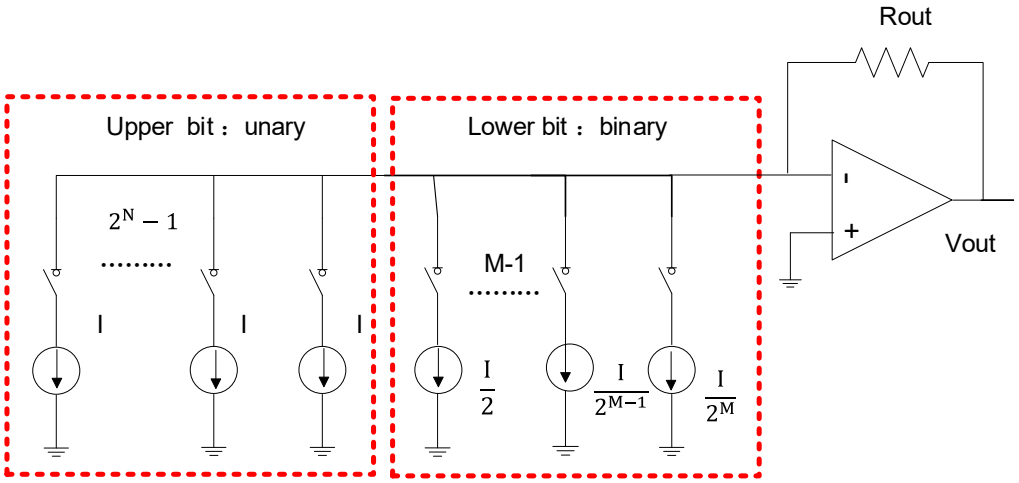


Fig. 1. Segmented DAC.

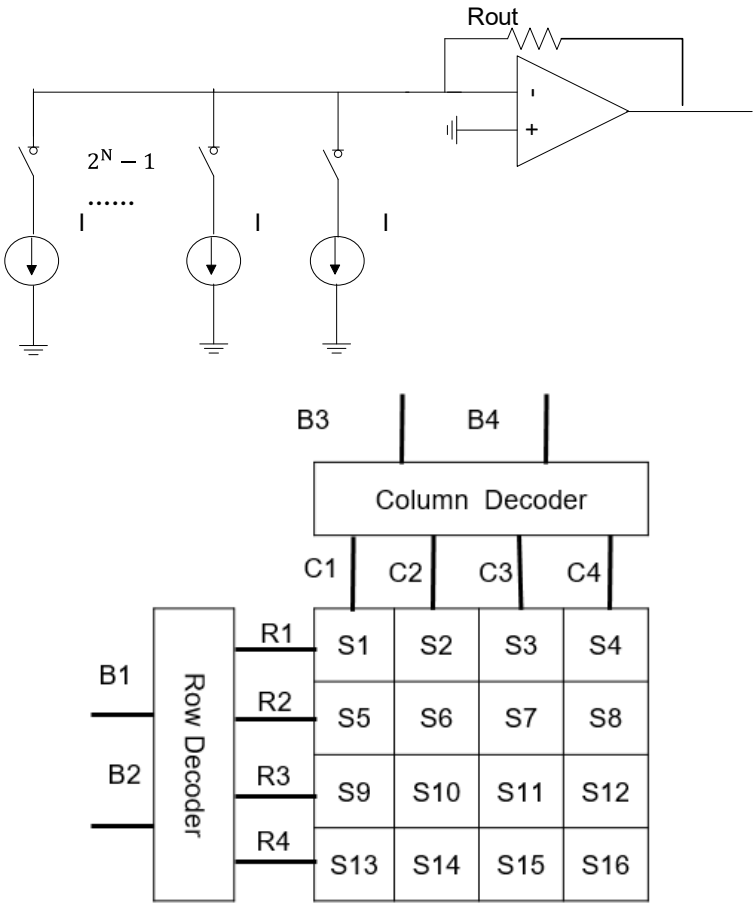


Fig. 2. Unary DAC circuit and regular layout of unit cell array. B4, B3, B2, B1 are binary inputs, while R1, R2, R3, R4, C1, C2, C3, C4 are their thermometer codes.

Many DACs are combination of unary and binary types. The unary type with low sensitivity devices is used for the higher bits, while the binary type with a small number of elements processes the lower bits. This yields a high-performance DAC with appropriate circuit size and power consumption.

3. Variations in Circuit Element Characteristics

There are systematic variations among MOSFETs, resistors and capacitors in integrated circuits, depending on their placement (location), and random variations which do not depend on placement. Ideally, the input and output characteristics of the DAC should be linear. However, in reality due to these variations, it can be nonlinear. The causes of these variations are as follows [2][9] ~ [13]:

1) Systematic variations.

a) Voltage drop on wiring.
b) Temperature distribution
c) CMOS manufacturing process
d) Doping distribution
e) Changes in threshold voltage
f) Accuracy in wafer plane
g) Mechanical stress

2) Random variation

a) Device characteristics mismatch.

The systematic variations can be modeled as linear and quadratic gradients as well as their combination, with regard to circuit element placement.

1) Linear gradient variation

- Voltage drop on wiring
- CMOS manufacturing process

2) Quadratic gradient variation

- Accuracy in wafer plane
- Temperature distribution
- Mechanical stress.

The above variations are significant determiners of the DAC linearity. The variations among the unit current sources 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

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

θ: angle of inclination g_l :magnitude of the slope

2) Quadratic error

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

g_q : variable value a_0 :position

3) Linear and quadratic joint errors

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

The influence of the systematic variation on the DAC linearity can be mitigated by technique to the target unit cell selection algorithm or layout. In the case of the segmented DAC, we consider that the variation effects may be reduced by adopting the algorithms of Magic square, Latin Square, Knight tour and Euler Knight tour.

4. Magic Square Unit Cell Selection Algorithm

A Magic square has the 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. Thus we investigated the selection algorithm of the unit cells to reduce the systematic variation effects and improve the DAC linearity (Fig. 3).

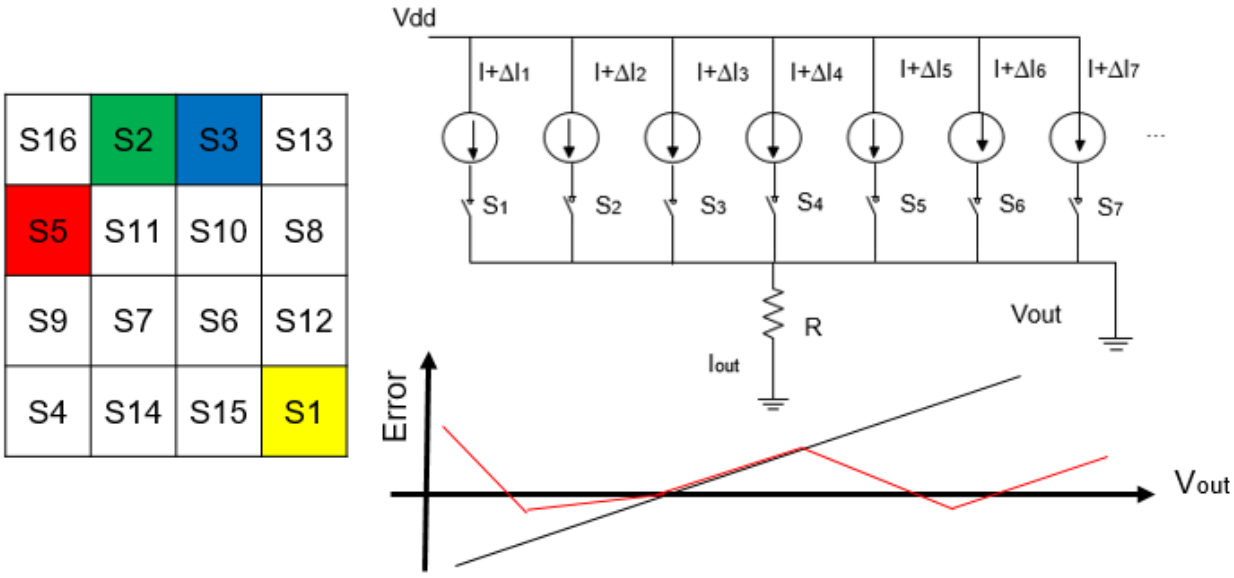


Fig. 3. Selection algorithm of unit cells for a unary DAC and its linearity improvement by cancelling linear gradient errors.

The Magic square for 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]. Given n elements on each row, column, and diagonal, the Magic square is usually called 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. 4, it can be confirmed that the sum of the elements of each row, column and diagonal components are all equal.

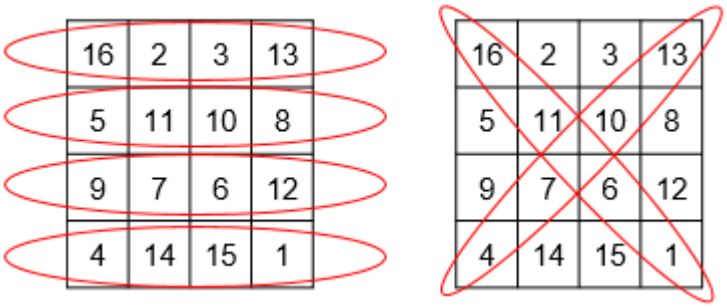


Fig. 4. Equivalent constant sum characteristics.

Utilizing this property, the systematic variations of a segmented DAC are expected to be reduced by the Magic square selection of the unit cells. The 16x16 Magic square used in the analysis is shown in Fig. 5.

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

Fig. 5. 16x16 Magic square.

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. 6) [6-8]. In a wide variety of Latin squares, a complete Latin square is defined such that for even n, numbers 1 through n in the first row are arranged in the following order : 1, 2, n, 3, n-1, 4, n-2, 5, n-3, …, n/2+2, n/2+1. For example, in case of n=16, we have 1, 2, 16, 3, 15, 4, 14, 5, 13, 6, 12, 7, 11, 8, 10, 9, and in case of n=4, we have 1, 2, 4, 3. In each remaining empty cell, place the number in the cell directly above it plus 1 by taking its modulo of n (Fig. 7). This was studied by mathematician Leonhard Euler (1707-1783).

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

Fig. 6. 4x4 Latin square.

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

Fig. 7. 4x4 complete Latin square.

6. Euler Knight Tour Unit Cell Selection Algorithm

The Knight tour is a sequence of moves that replicate Knight's moves on a chessboard such that the Knight piece visits every square exactly once; Fig. 8 shows where the Knight can move starting from around the center of an 8x8 board. If the Knight ends on a square that is one Knight's move from the beginning square so that it could tour the board again immediately, following the same path, the tour is closed; otherwise, it is open. Fig. 9 shows an example. The Euler Knight tour has both properties of Magic square and Knight tour as shown in Fig. 10.

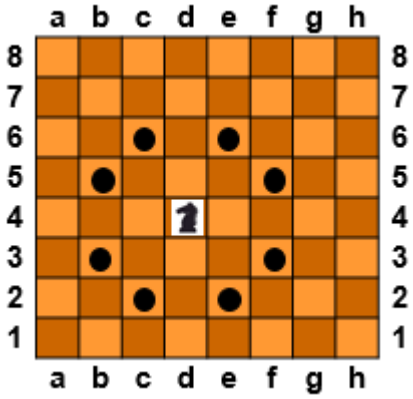


Fig. 8. Places where a Knight piece can move starting from around the center of an 8x8 chess board.

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

Fig.9. Knight tour algorithm in an 8x8 matrix.

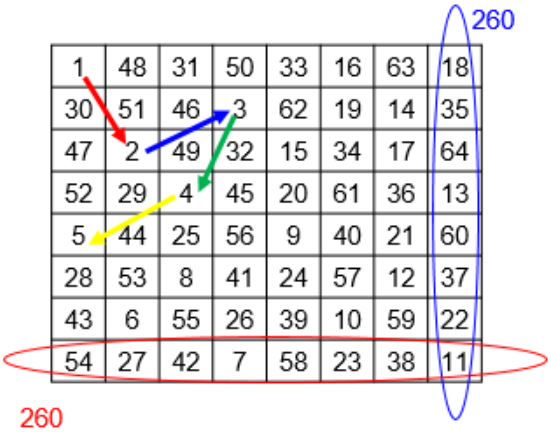


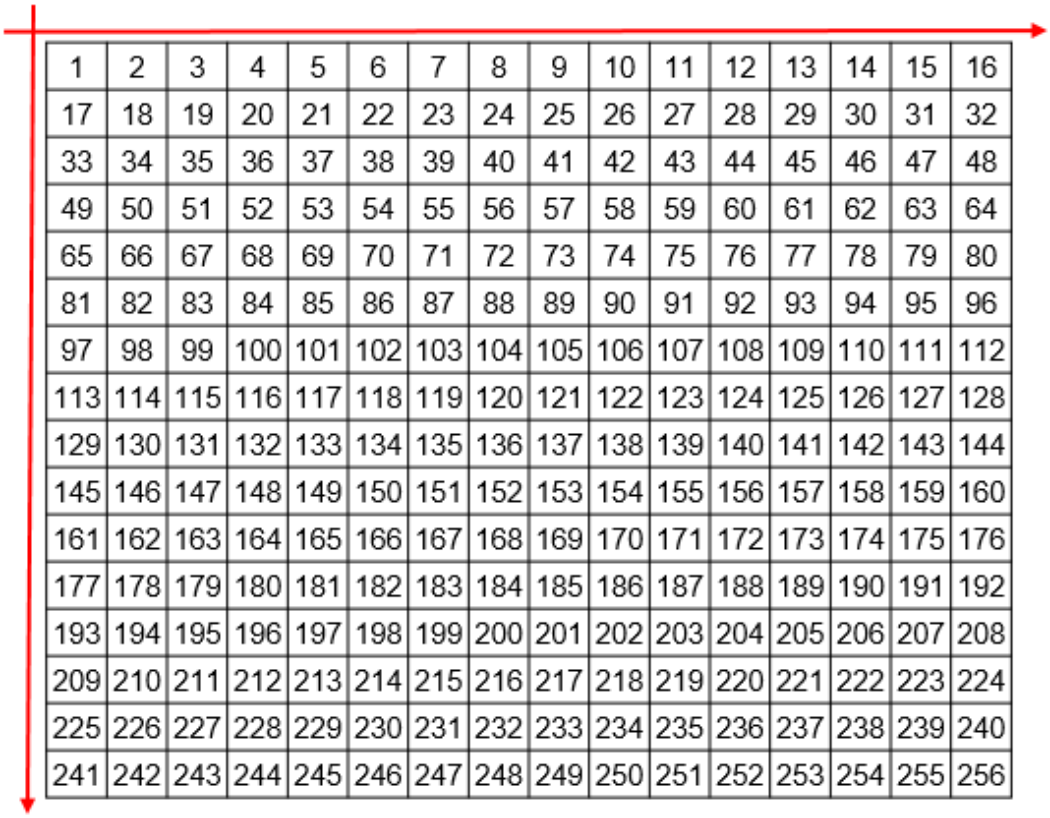
Fig.10. Euler Knight tour algorithm in an 8x8 matrix.

7. Simulation Results

The unary DAC with regular layout allows variations of the linear and quadratic gradients to degrade its linearity. Fig. 11 shows the case of 8-bit (16 × 16). We compared this regular layout, the Magic square layout (Fig. 12), the Latin square layout (Fig. 13) and the Euler Knight tour layout (Fig. 14).

We have simulated static performance of an 8-bit unary DAC, and compared the regular layout, the Magic square layout and the Euler Knight tour layout cases, and evaluated their performance using integral non-linearity (INL) for the case of linear gradient error (Fig. 15) and INL for the case of quadratic gradient error (Figs. 16 (a)(b)(c)(d)). The simulated frequency domain performance was assessed using the spurious free dynamic range (SFDR) in the case of linear gradient error (Figs.17-20); mismatch among unit cells was generated by setting a random number between -1 and +1. We see that the DAC with the Euler's Knight tour algorithm had superior SFDR in the given variation parameters.

The numerical simulation results in the linear gradient variation case are shown in Fig. 15 and the results in the quadratic gradient variation are shown in Figs. 16 (a) (b) (c) (d). We see that the Magic square, Latin square and the Euler Knight tour methods yield almost the same DAC 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. 11. Regular layout of 2D array current cells.

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

Fig. 12. Magic square layout of 2D array current cells.

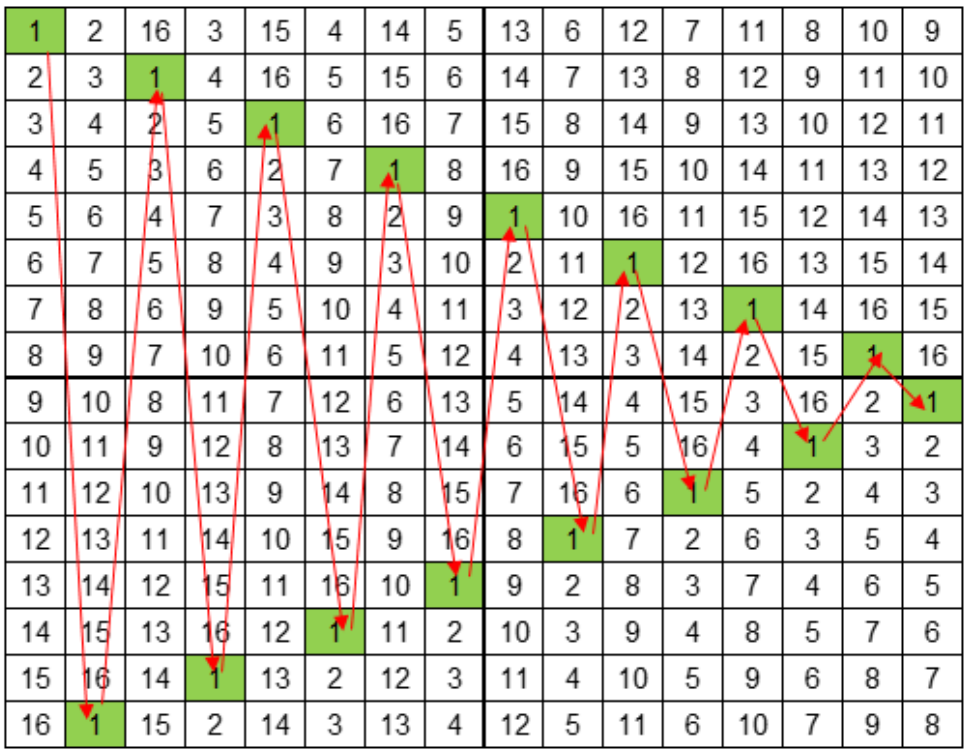


Fig. 13. Complete Latin square layout of 2D array of current cells.

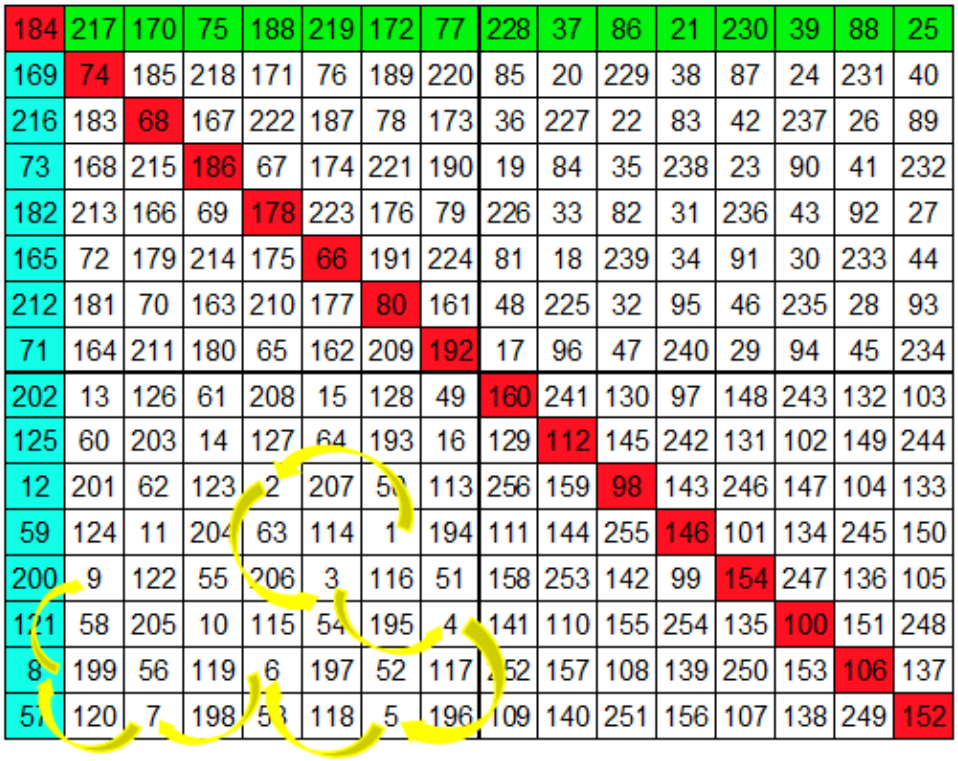


Fig. 14. Euler Knight tour layout of 2D array current cells.

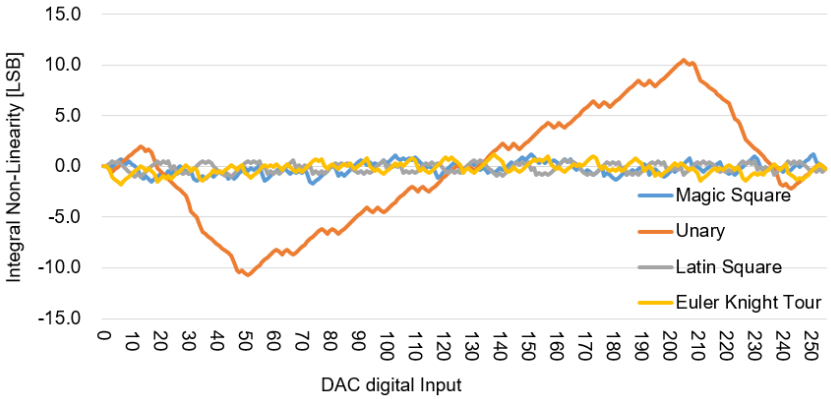
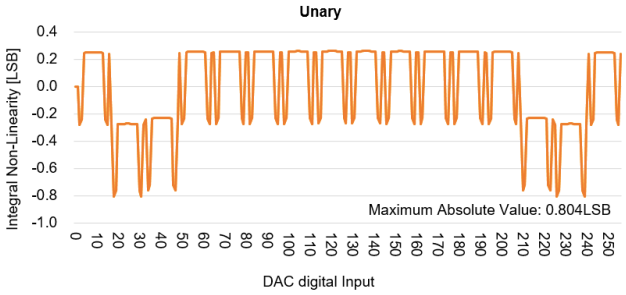
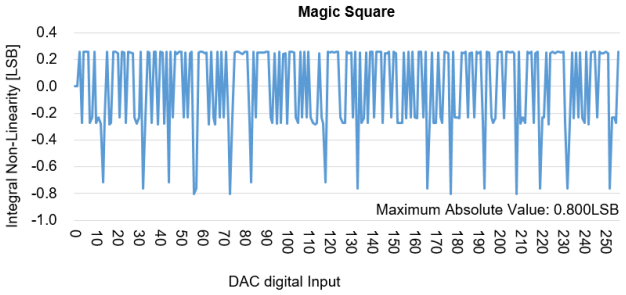


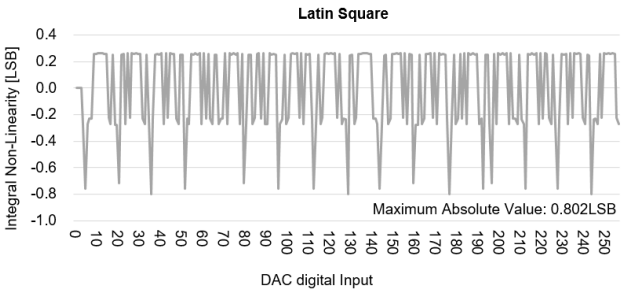
Fig. 15. Simulated INL in case of linear gradient.



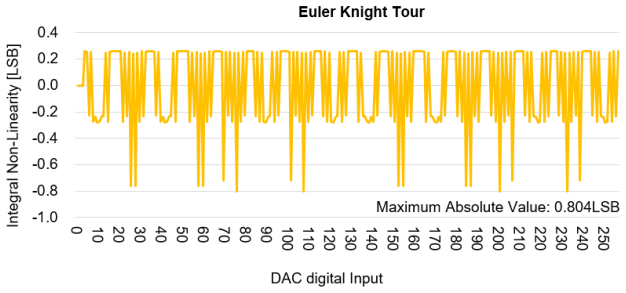
(a) Regular layout case



(b) Magic square case



(c) Latin square case



(d) Knight tour case

Fig. 16. Simulated INLs for quadratic gradient error.

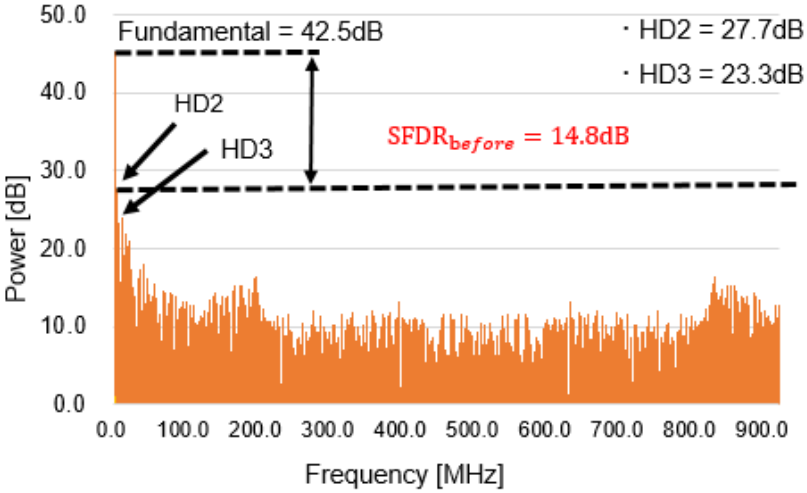


Fig. 17. SFDR for the regular layout.

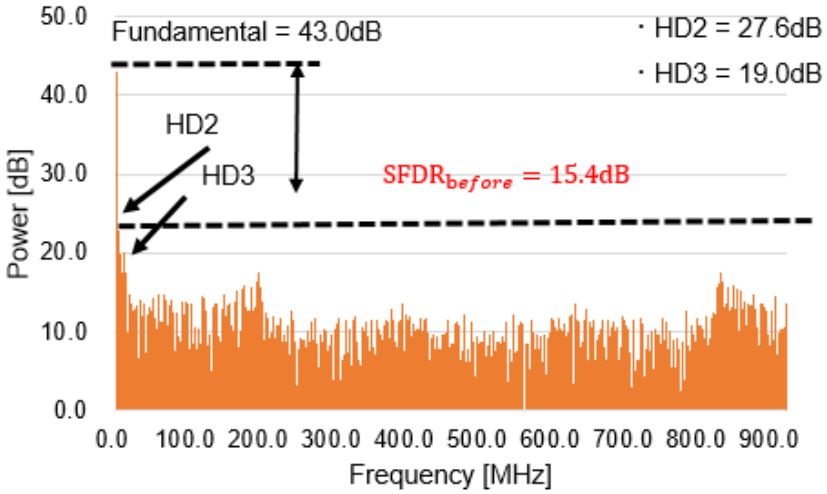


Fig. 18. SFDR for the Magic square layout.

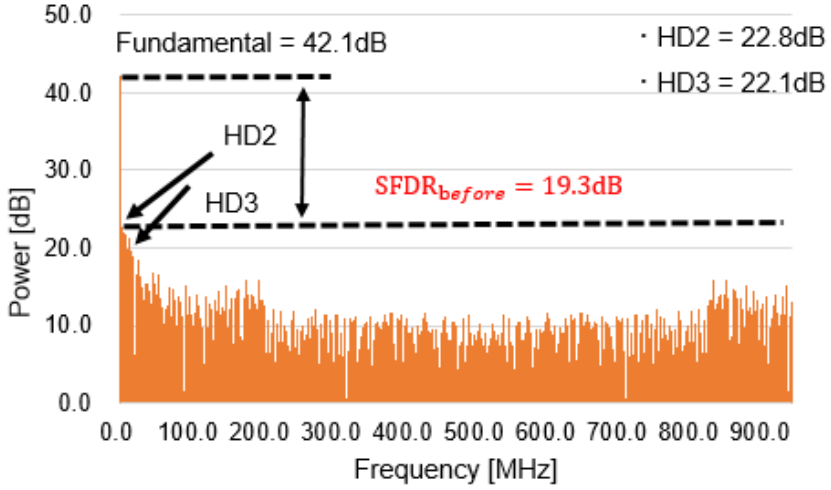


Fig. 19. Simulated SFDR for complete Latin square case.

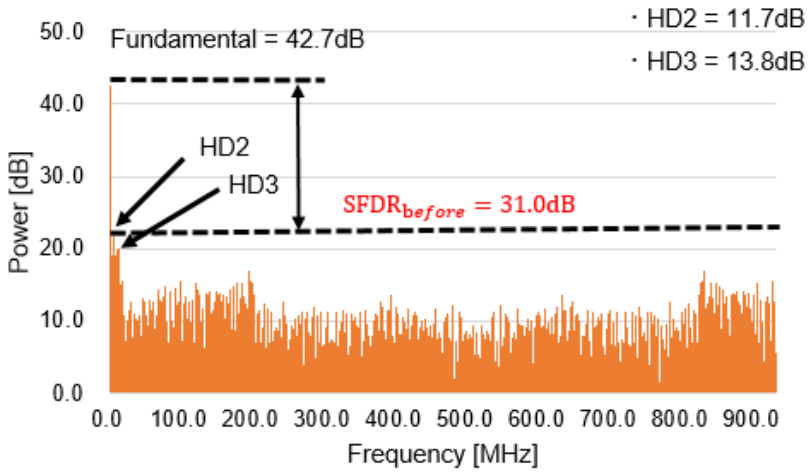


Fig. 20. Simulated SFDR for Euler Knight tour layout case.

The data simulation in one case may be contingent, so SFDR simulation was carried out according to the different data arrangement modes as shown in Figs. 21-23, and the results were obtained in Figs. 24-26. We see that when the variation obeys the linear distribution (Fig. 21), Latin Square is the best,

followed by Magic Square, Euler Knight Tour, and Regular. In the quadratic distribution case (Fig. 22), the Euler Knight tour is the best, followed by Latin Square, Magic Square, and then Regular. In the linear and quadratic combined distribution case (Fig. 23), the Latin Square and Euler Knight tour are the best followed by Magic Square and then Regular.

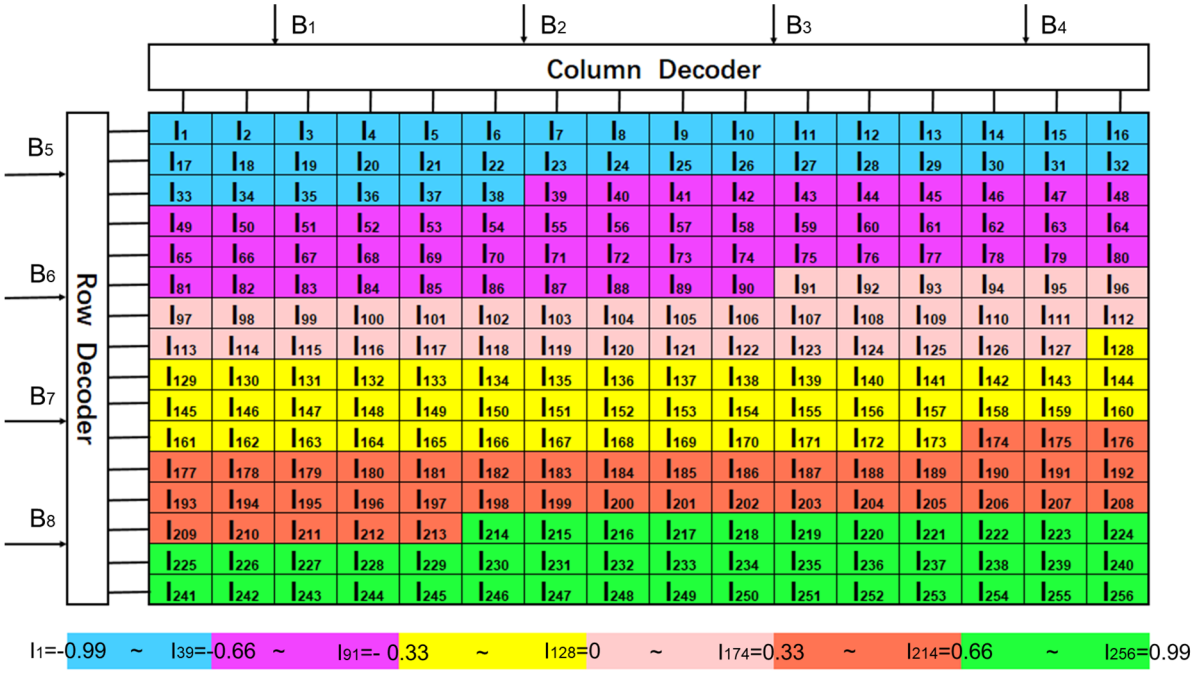
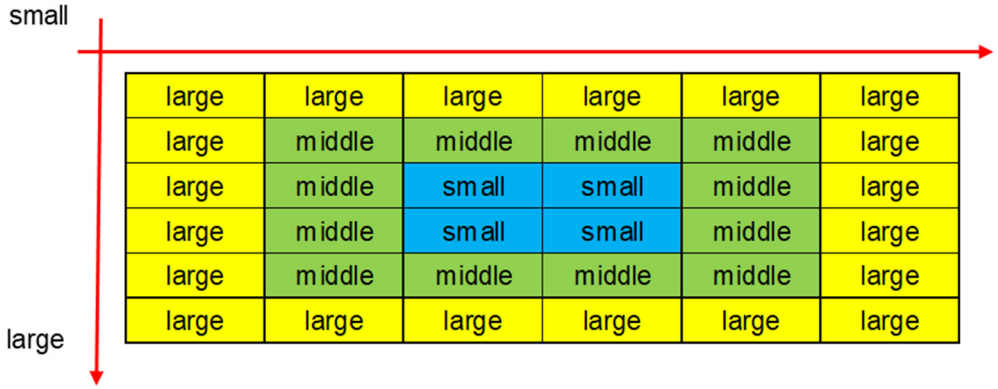


Fig. 21. Simulation condition is linear distribution of variation (Top to bottom, left to right, small to large).

large	large	large	large	large	large
large	middle	middle	middle	middle	large
large	middle	small	small	middle	large
large	middle	small	small	middle	large
large	middle	middle	middle	middle	large
large	large	large	large	large	large

Fig. 22. Simulation condition is quadratic distribution of variation (From the center to the periphery from small to large).



large	large	large	large	large	large
large	middle	middle	middle	middle	large
large	middle	small	small	middle	large
large	middle	small	small	middle	large
large	middle	middle	middle	middle	large
large	large	large	large	large	large

Fig. 23. Simulation condition is linear + quadratic distribution of variation (From the center to the periphery from small to large and top to bottom, Left to right, small to large).

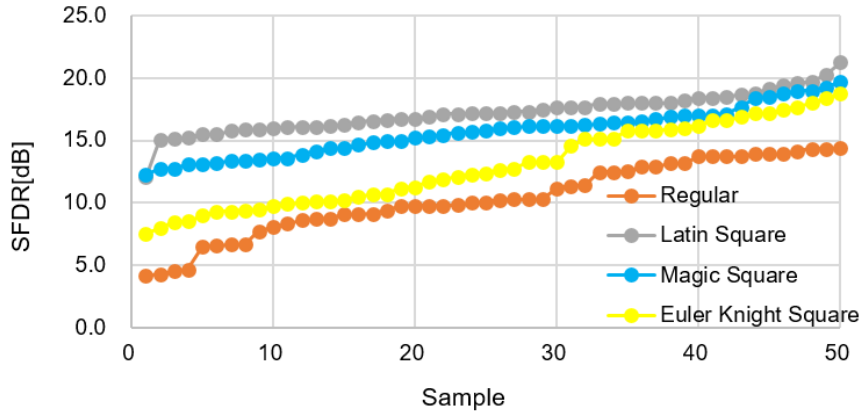


Fig. 24. Simulated SFDR result with linear variation using 50-Monte Carlo analysis.

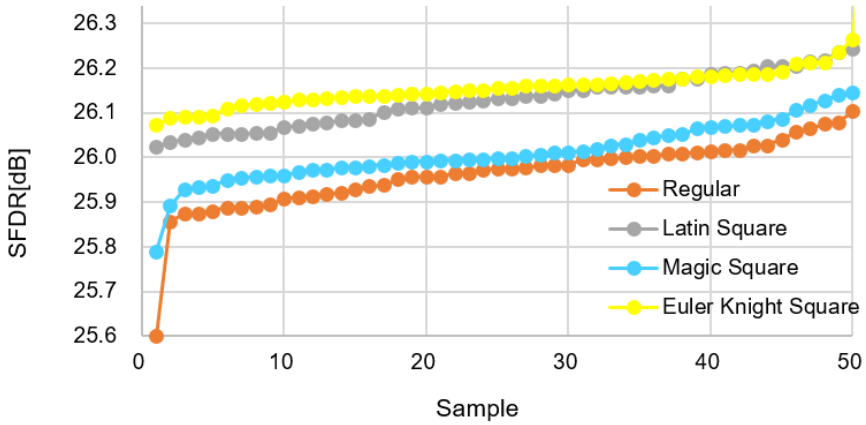


Fig. 25. Simulated SFDR result with quadratic variation using 50-Monte Carlo analysis.

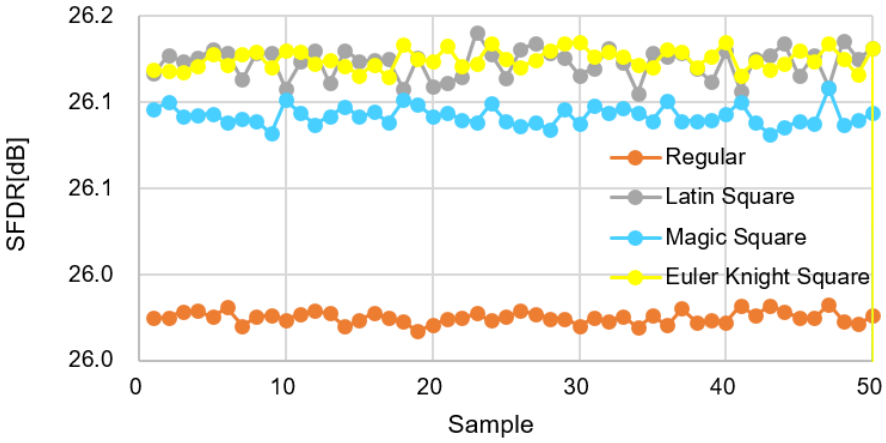


Fig. 26. Simulated SFDR result with linear+quadratic variation using 50-Monte Carlo analysis.

8. Layout and Routing Feasibility

We have developed a layout and routing design EDA tool for 2D unit cell arrays for regular, Magic square, Latin square and Euler Knight tour algorithms, to assess their feasibilities for DAC implementation on an IC because these algorithms are more complicated than the regular unit cell selection algorithm.

Fig. 27 shows the floor plan of the 2D unit cell array for the 6-bit (8x8) case while Fig. 28 shows the 2D regular layout case [9]; Fig. 29 shows the Euler Knight tour case. The binary digital input signals (B1, B2, B3, B4, B5, and B6) are fed into the column and row decoders simultaneously. The column decoder produces the thermometer code (C1, C2, …,C7) from B1, B2, B3 while the row decoder generate the thermometer code (R1, R2, …, R7) from B4, B5, B6. These thermometer codes are provided to each cell and by using each local decoder, the switch control signals S01 to S64 are generated as follows:

S01=R1+C1,	S02=R1+C2,	S03=R1+C3,	S04=R1+C4,
S05=R1+C5,	S06=R1+C6,	S07=R1+C7,	S08=R1,
S09=R2+R1*C1,	S10=R2+R1*C2,	S11=R2+R1*C3,	S12=R2+R1*C4,
S13=R2+R1*C5,	S14=R2+R1*C6,	S15=R2+R1*C7,	S16=R2,
S17=R3+R2*C1,	S18=R3+R2*C2,	S19=R3+R2*C3,	S20=R3+R2*C4,
S21=R3+R2*C5,	S22=R3+R2*C6,	S23=R3+R2*C7,	S24=R3,
S25=R4+R3*C1,	S26=R4+R3*C2,	S27=R4+R3*C3,	S28=R4+R3*C4,
S29=R4+R3*C5,	S30=R4+R3*C6,	S31=R4+R3*C7,	S32=R4,
S33=R5+R4*C1,	S34=R5+R4*C2,	S35=R5+R4*C3,	S36=R5+R4*C4,
S37=R5+R4*C5,	S38=R5+R4*C6,	S39=R5+R4*C7,	S40=R5,
S41=R6+R5*C1,	S42=R6+R5*C2,	S43=R6+R5*C3,	S44=R6+R5*C4,
S45=R6+R5*C5,	S46=R6+R5*C6,	S47=R6+R5*C7,	S48=R6,
S49=R7+R6*C1,	S50=R7+R6*C2,	S51=R7+R6*C3,	S52=R7+R6*C4,
S53=R7+R6*C5,	S54=R7+R6*C6,	S55=R7+R6*C7,	S56=R7,
S57=R7*C1,	S58=R7*C2,	S59=R7*C3,	S60=R7*C4,
S61=R7*C5,	S62=R7*C6,	S63=R7*C7,	S64=0.

As it is desirable that the digital and analog circuits/signals do not intersect each other, we put an output buffer on the right side of the DAC. The output terminal of all unit cells is connected to the input terminal of the output buffer cell.

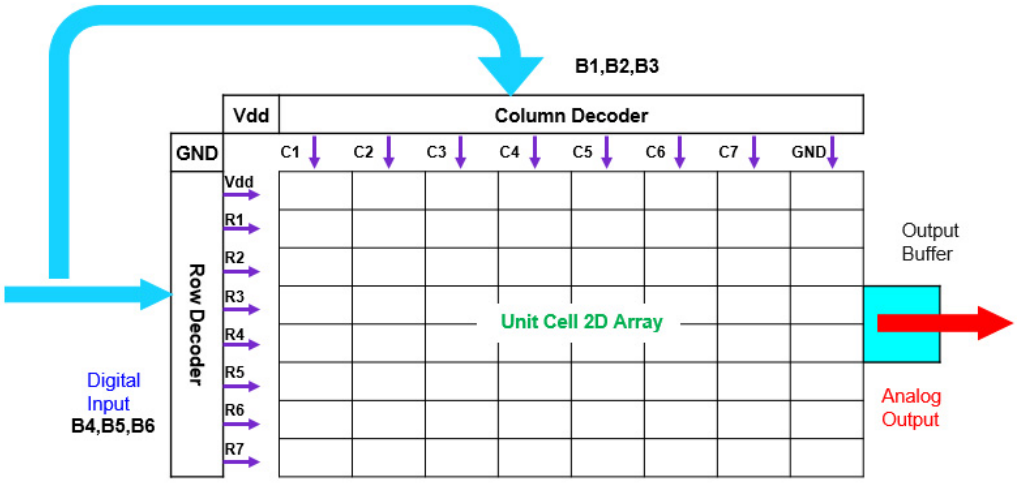


Fig. 27. 6-bit unary DAC floor plan.

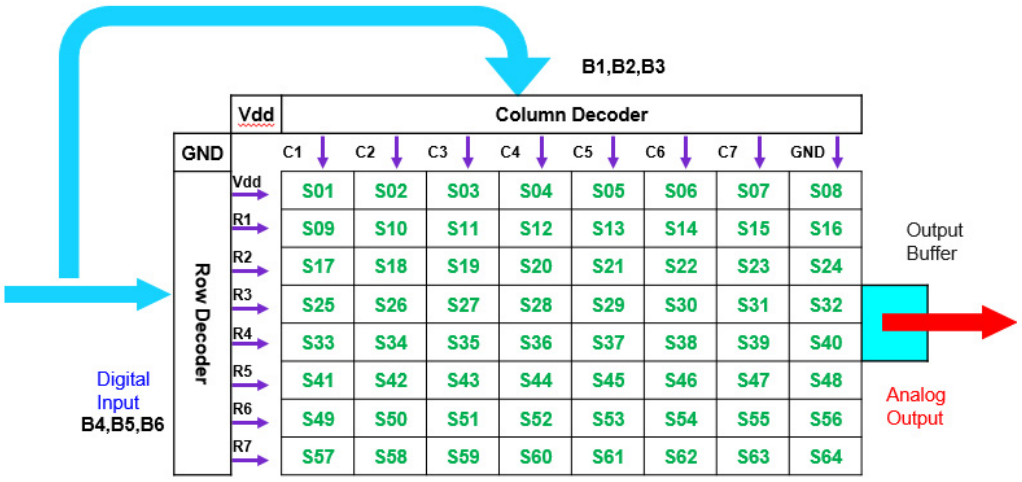


Fig. 28. Unary DAC floor plan for 2D regular unit cell array.

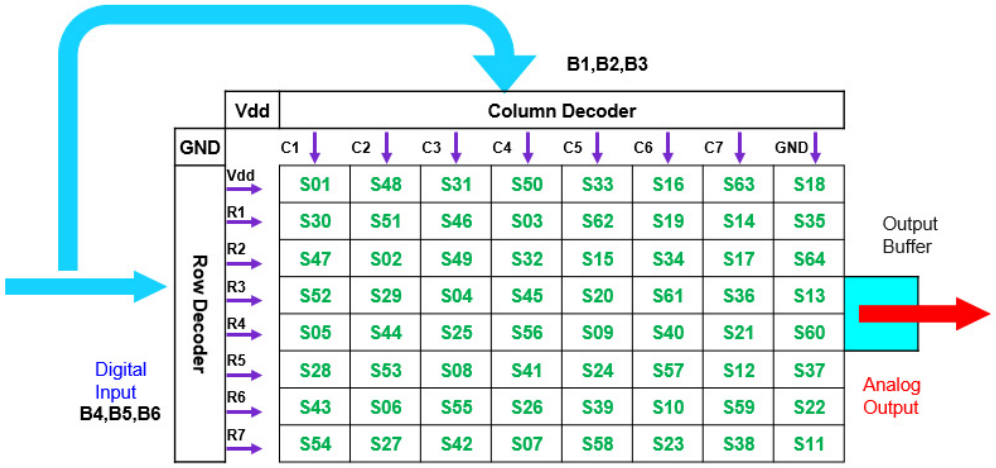


Fig. 29. Unary DAC floor plan for 2D unit cell array based on Euler Knight Tour.

Fig. 30-33 show the trial 2D layout and routing for 8-bit (16x16) unit cells in the regular, Magic square, Latin square and Euler Knight tour cases, respectively. We understand intuitively from these figures that while layout and routing are much more complicated the in Magic square, Latin square and Euler Knight tour cases, they are feasible for IC implementation.

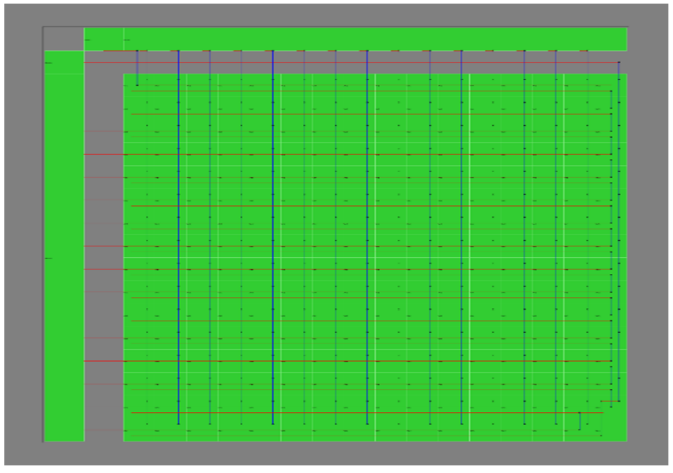


Fig. 30. 8-bit (16x16) unit cell 2D layout and routing based on Regular square.

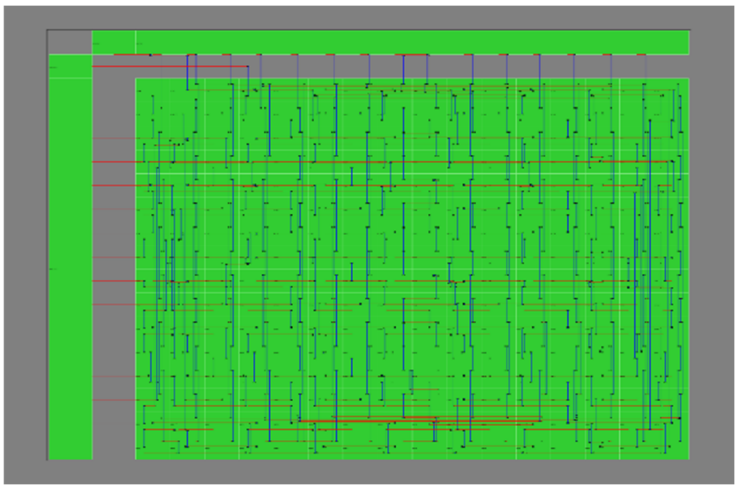


Fig. 31. 8-bit (16x16) unit cell 2D layout and routing based on Magic square.

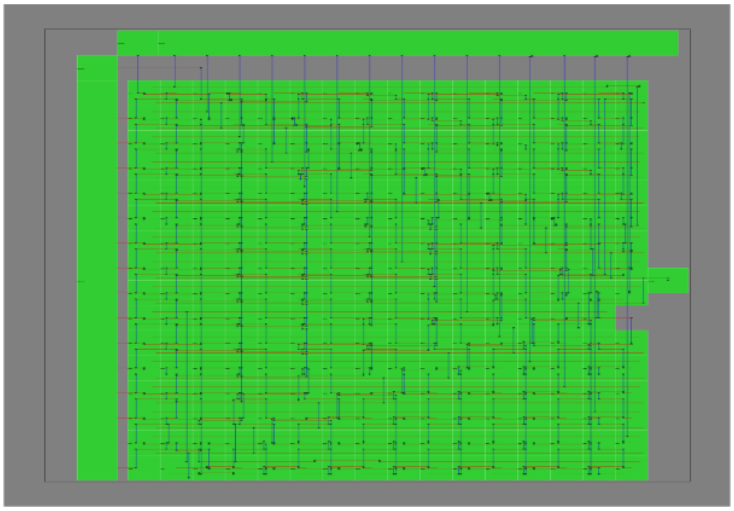


Fig. 32. 8-bit (16x16) unit cell 2D layout and routing based on Latin square.

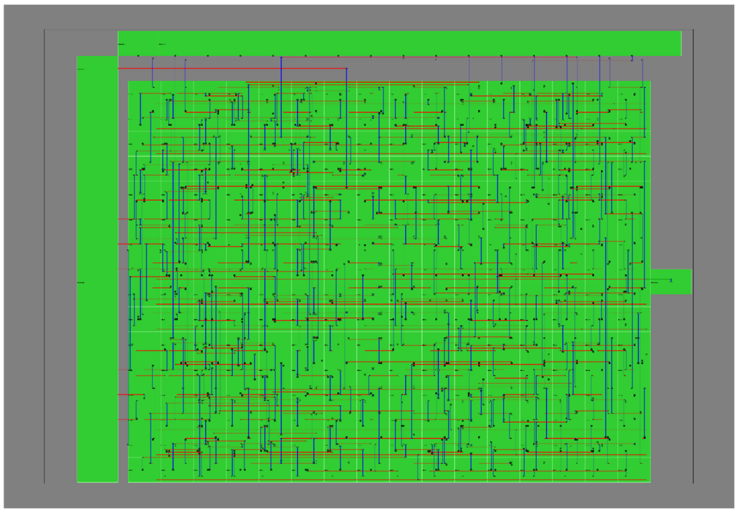


Fig.33. 8-bit (16x16) unit cell 2D layout and routing based on Euler Knight tour square.

9. Conclusion

In this paper, we have demonstrated that unary cell selection algorithms based on the classical mathematics can improve the segmented DAC linearity as they can cancel systematic mismatch effects. Pseudo random numbers in 2D arrays were simulated to reproduce by the cell arrangements using the regular, Latin square, Magic square and Euler Knight tour approaches. Our simulations showed that Latin square, Magic square and Euler Knight tour algorithms offer superior overall DAC linearity to the regular layout. We have also developed a layout and routing design EDA tool for the 2D unit cell arrays for regular, Latin square, Magic square and Euler Knight tour algorithms, and shown their feasibilities for DAC implementation in an IC.

We expect that since there are a lot of Euler Knight tour, Latin square and Magic square mathematical research results, layout algorithms using them to configure 2D unit cells of the unary DAC can be refined further.

Acknowledgements

This work is supported by Adaptable and Seamless Technology Transfer Program through Target-Driven R&D (A-STEP) from Japan Science and Technology Agency (JST) Grant Number JPMJTR201C.

Data Availability

Data sharing will be made available with reasonable request.

References

[1] ***Data Converters*****,** F. Maloberti, Springer New York (New York, USA), 2007.

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

[3] M.J.M. Pelgrom, AAD C.J. Duinmaijer, and A.O.G. Welbers, "Matching Properties of MOS Transistors", ***IEEE Journal of Solid-State Circuits,*** Vol.24, No.5, pp.1433-1440, 1989.

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

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

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

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

[8] W.L. Stevens, "The Completely Orthogonalized Latin Square", ***Annals of Human Genetics, Cambridge University Press,*** Vol.9, No.1, pp. 82-93, 1939.

[9] T. Miki, Y. Nakamura, M. Nakaya, S. Asai, Y. Akasaka, and Y. Horiba, "An 80-MHz 8-bit CMOS D/A Converter", ***IEEE Journal of Solid-State Circuits,*** Vol.21, No.6, pp.983-988, 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, 1999.

[11] Y. Cong and 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, 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, 2011.

[13] X. Li, F. Qiao, and H. Yang,"Balanced Switching Schemes for Gradient-Error Compensation in Current-Steering DACs", ***IEICE Trans. Electron***, Vol.E95-C, No.11, pp.1790-1798, 2012.

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

[15] D. Yao, Y. Sun, M. Higashino, S. N. Mohyar, T. Yanagida, T. Arafune, N. Tsukiji, and H. Kobayashi, "DAC Linearity Improvement with Layout Technique Using Magic and Latin Squares", ***Proceedings of IEEE International Symposium on Intelligent Signal Processing and Communication Systems*** (Xiamen, China) November 2017.

[16] M. Higashino, S.N.B. Mohyar, Y. Dan, Y. Sun, A. Kuwana, and H. Kobayashi, "Digital-to-Analog Converter Layout Technique and Unit Cell Sorting Algorithm for Linearity Improvement Based on Magic Square", ***Journal of Technology and Social Science,*** Vol.4, No.1, pp.22-35, 2020.

[17] M. S. Yenuchenk, A.S. Korotkov, D.V. Morozov, and M.M. Pilipko, "A Switching Sequence for Unary Digital-to-Analog Converters Based on a Knight's Tour", ***IEEE Transactions on Circuits and Systems I***, Vol.66, No.6, pp.2230-2239, 2019.