DAC Linearity Improvement Algorithms Using Randomization Methods

DAN YAO

PhD Dissertation

DIVISION OF ELECTRONICS & INFORMATICS
GRADUATE SCHOOL OF SCIENCE & TECHNOLOGY
GUNMA UNIVERSITY

JAPAN

March 2023

DAC Linearity Improvement Algorithms

Using Randomization Methods

DISSERTATION

Submitted by

DAN YAO

In partial fulfillment of the requirements for the award of the Degree of

DOCTOR OF PHILOSOPHY

IN

ELECTRONICS & INFORMATICS ENGINEERING

Under the guidance of

PROFESSOR HARUO KOBAYASHI, Ph. D. Eng.

DIVISION OF ELECTRONICS & INFORMATICS

GRADUATE SCHOOL OF SCIENCE & TECHNOLOGY

GUNMA UNIVERSITY

JAPAN

March 2023

Acknowledgement

As I approach the end of my Ph.D. research, I would like to thank all those who have helped me in my life and studies at this important time in my life.

I would like to express my deepest gratitude to Professor Haruo Kobayashi for his encouraging guidance and encouragement in promoting this research. I was involved in research on improving the linearity of the DAC, and received many opportunities. Thanks to assistant professor Anna Kuwana, I receive the professor Anna Kuwana of research many times and thank you. I thank you from the bottom of my heart for becoming extremely helpful and became a major guideline for research. I would like to thank my advisers Professor Kuniyuki Motojima, Professor Toshiki Takahashi, Professor Yuji Gendai and Professor Masahiro Ishida for them valuable comments on my dissertation.

Thanks to the students Xueyan Bai, Yujie Zhao and Shogo Katayama who graduated with me for their encouragement and encouragement in writing the dissertation. I would like to thank Yu Cao of Hashimoto Lab for his great help in my research and valuable suggestions when I was confused. Thank the students in Kobayashi Lab research room for their help in my Japanese and help me when I have Japanese problems.

Thanks to my father, Chunbo Yao, and my mother, Cuimei Guan, who have been silently supporting and encouraging me behind me. It is my strong backing when I encounter problems.

Thanks for Mr. Kazuyuki Kawauchi and Jedat Inc. People

Declaration

I hereby undertake: This dissertation is my own research work.

There is no reproduction of the results in any previously published documents or articles. Dissertation is done under the scientific guidance of Prof. HARUO KOBAYASHI. The data, images, and research results presented in the dissertation are completely honest. Dissertation has references and uses materials posted on conferences, magazines, articles, websites, textbooks. All materials used in the dissertation are mentioned in detail in the references section.

I take full responsibility for the above guarantees.

Signature:

Name: YAO DAN

Student No.: T202D005

Date:

Abstract

The digital-to-analog converter (DAC) is a key component in electronic measurement instruments, communication systems and audio systems, and many applications require highly linear DAC. However, the DAC circuit suffers from manufacturing variations, which degrades the DAC linearity. In this dissertation, two technologies using randomization methods are proposed to reduce the manufacturing variation effects and improve the linearity of the unary DAC. The first one is unit cell selection algorithms in a pseudo random manner based on Magic Square, Latin Square and Euler Knight Tour. This is a static randomization method. It is shown in numerical simulations that these can reduce the systematic mismatch effects among unit cells and improve the DAC linearity. Also the layout and routing EDA tools are developed for these selection algorithms and it is shown that they are feasible for IC implementation. The second one is the method of scrambling both systematic and random mismatches among unit cells dynamically. If it is implemented directly for 2D unit cell array layout, relatively complicated circuit and routing are required. Here the scrambling circuit for virtual 3D unit cell array layout (but actually 1D layout) is considered and it is shown that its circuit and routing are relatively simple. Simulation verifications for the linearity improvement are also demonstrated.

Contents

List of Figures

List of Tables

Comparison Table of Abbreviations

Abbreviation	Full Name in English
DAC	Digital to Analog Converter
ADC	Analog to Digital Converter
INL	Integral Non-linearity
DNL	Differential Non-linearity
SNR	Signal to Noise Ratio
SNDR	Signal to Noise Distortion Ratio
SFDR	Spurious-free dynamic range
SINAD	Signal to Noise and Distortion Ratio
ENOB	Effective number of bits
DR	Dynamic Range
LSB	Least Significant Bit
MSB	Most Significant Bit

Chapter 1

Introduction

1.1 Research Background and Motivation

With the development of digital signal processing and digital computing technology, we are now enjoying life in the "digital" world more and more [1, 2]. Compared to analog circuits, using digital circuits is less affected by noise and more reliable. The advantage is that it is easier to integrate on a chip to realize complex functions. However, the signals touched in the real world are analog signals such as sounds and images. Therefore, conversion between analog signals and digital circuit signals is important. It is necessary to be able to easily convert an analog signal into a digital signal as an interface between them. The circuit that realizes this function is an analog-to-digital converter (ADC) [3].

ADCs are important modules in the configuration of electronic systems and often have a significant impact on the overall performance of the system. With the development of ultra-deep submicron CMOS processes, the degree of digital circuit integration increases [4, 5]. The price is getting higher and the function to realize it is complicated while the high-speed processing of signals and the demand for accuracy are getting higher. However, the development of ADC design is relatively slow, and the development of analog design software is immature. The development of analog interface

circuits lags behind the development of digital circuits. Especially in digital television video systems and digital communication systems, their performance (e. g., speed and accuracy) often limits the improvement of overall system performance [6,7].

Since the introduction of ADC, we have experienced the development of data conversion of discrete semiconductors and integrated circuits, and the development strategy of high-speed and high-precision ADC is to increase the degree of integration as much as possible and solve the product for the final user, assuming that the performance is not affected. It has gone through the process of semiconductor, integrated circuit data conversion to provide a solution. Nowadays, the demand for ADC is increasing significantly, and the performance index is wider in order to adapt to the requirements applied to various applications. It is required to be covered. The main fields of application of ADCs are constantly expanding, and they are widely applied in fields such as sensor, DSP, multimedia, communication, and measurement [8, 9]. ADCs meet the different demands of various fields. On the other hand, at the design stage, not only the process and circuit configuration of the ADC itself, but also the signal modulation is supported, and the peripheral circuit design of the ADC such as an analog circuit such as an analog filter is also performed. These should be taken into consideration.

1.2 Development History and Research Comparison of DAC

The development a digital-to-analog converter (DAC) has gone through the process of electron tube, transistor and integrated circuit. It is produced and developed because of the urgent needs of human production and life, and every stage of its development is related to the latest scientific and technological achievements at that time.

In the late 1940s, people began the research and practice of digital communication, and developed ADC and DAC mounted by the electron tube. With the development and maturity of transistor technology, to the late 1950s, the converter in the electron tube gradually replaced by the transistor, so that the volume and weight of the converter is greatly reduced. In the mid-1960s, the main functional unit circuit of the converter, such as operational amplifier, reference voltage source, resistance network, analog electronic switch and logic control circuit, has been gradually integrated, especially the integrated operational amplifier unit has entered the stage of large-scale industrial production. In the early 1970s, a single chip DAC with all components integrated on a single chip was developed. It marks the DAC has really reached the stage of industrial mass production, get rid of the trouble of carefully selecting the components of the converter, thus greatly reducing the cost and improving the reliability.

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 [5-10]; 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 [15-17] and a Russian group [18]. The proposed algorithm is expected to refine the DAC linearity improvement algorithm further according to the theoretical results for Magic Square and Knight Tour.

We remark that in our previous publication [15], the Magic Square algorithm is applied to sorting of the unit cells to improve the unary DAC linearity. However, this dissertation here describes unit cell selection algorithms based on the Euler's Knight Tour which has both properties of Knight Tour and Magic Square. Hence, these look similar but they are different.

The Knight Tour algorithm is expected to select the next unit cell close to the present cell so that the differential non-linearity (DNL) variation within adjacent digital inputs would not be large; if the next selected cell is far from the present cell, the DNL variation within adjacent digital inputs can be large due to systematic mismatch. The Magic Square algorithm, on the other hand, offers well-balanced cell selection [19]. 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 [15-17] 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 [18], but not Euler Knight Tour algorithm, even though Leonhard Euler stayed at Saint Petersburg for a long time. This dissertation 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 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 [10]. Subsequently, there have been several studies in this area [11-18]. Our proposal targets better 2D pseudo randomization, though layout and routing are complicated compared to [10].

1.3 Organization

In this dissertation, through the analysis and modeling of the structure, static and dynamic characteristics of the segmented DAC, the key issues of

DAC design are clarified, and the corresponding design techniques are discussed.

In the design, this dissertation rationally selects the segmented structure, proposes a simplified solution method for row-column decoding, optimizes the arrangement of the current source array, and makes the designed segmented DAC achieve better static and dynamic characteristics.

The dissertation is divided into five chapters.

Chapter 1 introduces research background for importance of DAC linearity improvement as well as the research motivation and purpose. Also, our approaches are introduced and possible applications of the linear segmented DAC are described.

Chapter 2 introduces basic circuit structure and operation principle of the unary DAC. Also, performance criteria of DAC characteristics and definitions of various characteristic parameters are explained. Error sources for mismatches among unit cells of the unary DAC are studied.

In Chapter 3, various unit cell selection algorithms of the unary DAC for 2D layout are studied. Mathematical characteristics of Magic Square, Latin Square and Euler Knight Tour are introduced. Magic Square has the balance of rows, columns and diagonals, and Latin Square has the balance of each unit, while Euler Knight Tour has both attributes of Knight Tour and Magic Square. Cell selection algorithms in a static pseudo random manner based on Magic Square, Latin Square and Euler Knight Tour are proposed to

improve the linearity of the unary DAC by cancelling system mismatch effects among unit cells. Simulation verifications have shown that they can improve the linearity of the unary DAC with the practical assumption of linear, quadratic or linear/quadratic commination systematic mismatches. The effectiveness comparison for the three algorithms as well as the regular conventional algorithm is also shown in simulations. We have also developed a layout and routing design EDA tool for the 2D unit cell arrays for regular, Magic Square, Latin Square, and Euler Knight Tour algorithms, and shown their feasibilities for DAC implementation in an IC.

Chapter 4 studies a unit cell mismatch scrambling method for a high-resolution unary DAC based on virtual 3D layout. DAC for communication applications requires good spurious free dynamic range (SFDR) performance. The SFDR of the unary DAC is degraded due to static characteristics mismatches among its unit cells as well as its dynamic characteristics. In this dissertation, we focus on the care for the static characteristic mismatches using the dynamic element matching (DEM) for the unary DAC. We consider here the signal bandwidth of the DAC is from DC to the Nyquist rate (half of the sampling frequency) so that the DEM technique only needs to spread the spectrum of the spurious tones caused by the mismatches in the entire signal band uniformly without need for mismatch shaping; this is called as mismatch scrambling. However, direct implementation circuit of such mismatch scrambling for the high-resolution unary DAC becomes complicated. Therefore, first in simulations, a

mismatch scrambling technique utilizing the features of 2D regular layout and routing for the unit cells was investigated, utilizing row and column decoders features and this method can reduce the hardware compared to the direct method. Then, we have investigated a more hardware efficient method for a higher resolution DAC, by considering a virtual 3D case, considering X-,Y- and Z-decoders, but there 1D layout is actually used. Its circuit implementation requires only small amount of circuits compared to the conventional methods. Simulation results show that its SFDR can be improved compared to the case without the proposed scrambling method.

Chapter 5 summarizes the research work of this dissertation as well as describes the future work.

References

[1] Yonghua Cong, Randall L. Geiger. “A 1.5-V 14-Bit 100-MS/s Self-Calibrated DAC”. IEEE Journal of Solid-State Circuits, vol. 38, no. 12 (Dec. 2003.

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

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

[4] 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).

[5] K. Omori, The World of Magic Square, Nippon Hyoron Sha,

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

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

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

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

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

[11] 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).

[12] 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)

[13] 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).

[14] 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).

[15] 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).

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

[17] M. Higashino, S. N. B. Mohyar, Y. Dan, Y. Sun, A. Kuwana, 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 (Jan. 2020).

[18] M. S. Yenuchenk, A. S. Korotkov; D. V. Morozov, 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 (June 2019).

Chapter 2

DAC Main Structure and Performance Index

In order to communicate the digital signal processor with the analog world, data acquisition and reconstruction circuits are required: analog-to digital-conversion takes and digitizes the front-end analog signal, and digital-to-analog conversion processes the signal on the back-end Fig. 2.1 is the block diagram of the converter system.

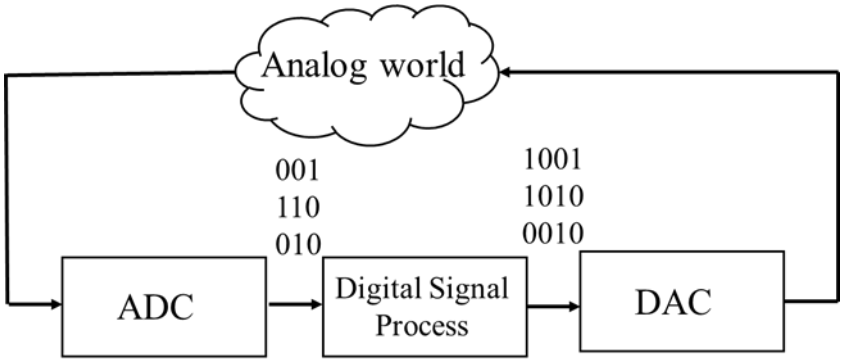


Fig. 2.1 Digital processors and analog world interface systems.

2.1 Basic Principle of DAC

The function of the DAC is to convert the digital signal into an analog signal with continuous time and discrete amplitude [19]. Fig.2.2 is a schematic diagram of the structure and composition of a DAC. A complete DAC typically consists of a reference voltage source, weighted network, analog switches, and an output amplifier.

Among them, the reference voltage source provides a reference voltage for the DAC, and is generally composed of a bandgap reference circuit and an operational amplifier with a strong ability to suppress temperature and power supply. The weighted network enables each bit of the DAC to generate a voltage or current with a certain weight, each bit of the unary DAC has the same weight, and the relationship between the adjacent bit weights of the binary DAC is binary. The analog switch controls the transmission of each bit of data in the DAC. Generally, NMOS transistor switch, PMOS transistor switch or CMOS switch is selected. By controlling the gate voltage of the MOS transistor, the on-off of the analog switch can be controlled [20]. Usually, an operational amplifier is designed at the output end of the DAC, so that the DAC has a certain driving capability [21].

The basic working principle of the DAC is to add the digital codes according to their corresponding weights to generate a linear analog voltage or current. Taking N-bit binary DAC as an example, if the digital code input by the DAC is $\mathrm{D}(d_{N-1}, d_{N-2}, \ldots, d_3, d_2, d_1)$, the output voltage can be expressed as:

$$V_{OUT} = V_{ref}\left(\frac{d_{N-1}}{2^1} + \frac{d_{N-2}}{2^2} + \cdots + \frac{d_3}{2^{N-2}} + \frac{d_2}{2^{N-1}} + \frac{d_1}{2^N}\right) \qquad (2\text{-}1)$$

In Eq. (2-1), Vout represents the output voltage of the DAC and Vref represents the reference voltage. It can be seen from the expression that the weight value of each bit of the DAC decreases bit by bit from the most

significant bit (MSB) to the binary, and the weight value of the least significant bit (LSB) is the smallest, $1/2^N$.

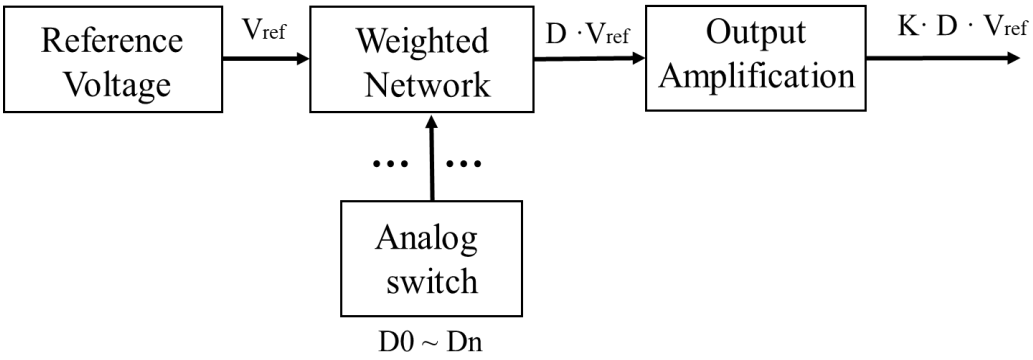


Fig. 2.2 Schematic diagram of DAC.

2.2 Basic Structure of DAC

There are many ways to classify DACs. Many literatures classify them according to the basic structure of DACs. The DAC with the basic structure of voltage divider network is called voltage scaling DAC, the DAC with the basic structure of shunt network is called current scaling DAC, and the DAC with the basic structure of capacitive storage array is called charge scaling DAC [22]. It can also be classified according to the basic unit form of the DAC. For example, the DAC whose basic unit is the current source is called the current steering DAC. In addition, the DAC can also be divided into the serial DAC and the parallel DAC according to the data transmission mode of the DAC digital input.

This dissertation classifies the basic structures of DACs into unary DACs, binary DACs, and segmented DACs combined with multiple

structures according to the binary relationship of the adjacent bits of the DACs. This classification method starts from the overall structure of the DAC and does not stick to the form of the basic unit, making the basic classification of the DAC clearer. Generally speaking, unary DAC, binary DAC and segmented DAC include two versions of voltage output mode and current output mode.

2.2.1 String DAC and Unary DAC (Thermometer DAC)

The main structure of the string DAC is composed of a string of resistors with the same resistance value [23]. An 3-bit string DAC circuit structure diagram, as shown in Fig. 2.3. We see that when the lowest bit switch is turned on, the output voltage is zero, and when the highest bit switch is turned on, the output voltage is 7Vref/8.

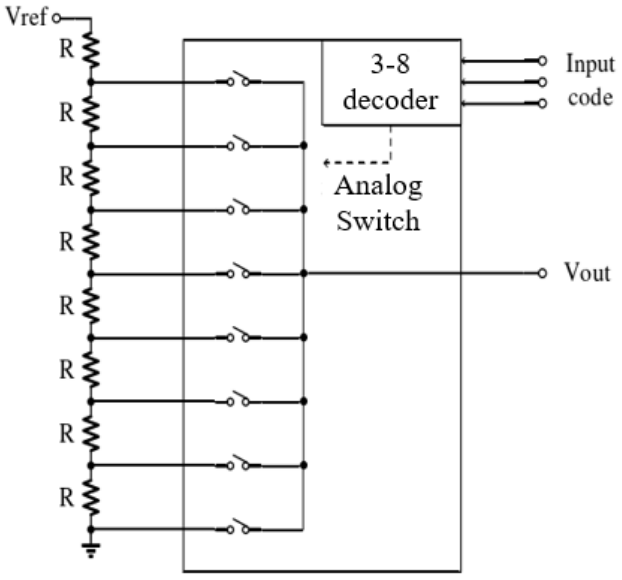


Fig. 2.3 Voltage output string DAC.

String DAC itself has monotonicity. Even if a resistor in the resistor string is mismatched or short-circuited, the output of the n-th bit will not be greater than the (n+1)-th bit. This is an advantage of the string DAC. Since only one switch is turned on during the operation of string DAC, no more than two switches can operate during one state switching, which makes the switching burr of string DAC relatively constant and does not contribute much to the harmonic of the output signal of the log analog converter.

The number of resistors in the N-bit string DAC is 2^N. For low-to-medium resolution DACs, this structure is feasible, but for high-resolution DACs, the string DAC structure requires a huge translation code circuits and a large number of resistors and switches, The increase in the number of resistors increases the probability of resistor mismatch, resulting in a decrease in the static parameters of the DAC, which limits the application of the string DAC structure in high-resolution DACs [24].

Unary DAC can also be called a thermometer DAC or a fully decoded DAC. Fig. 2.4 is a circuit structure. Each resistor in the unary DAC structure is connected in parallel. One end of the resistor is connected to a reference voltage source, and the other end is connected to a switch. This structure still has monotonicity, that is, the number of thermometer DAC switches turned on can only change from less to more or from more to less, and the output voltage of the DAC can only change from low to high or from high to low accordingly. Thermometer DACs suffer from the same problems as string DACs, and the unary structure limits their use in high-resolution DACs.

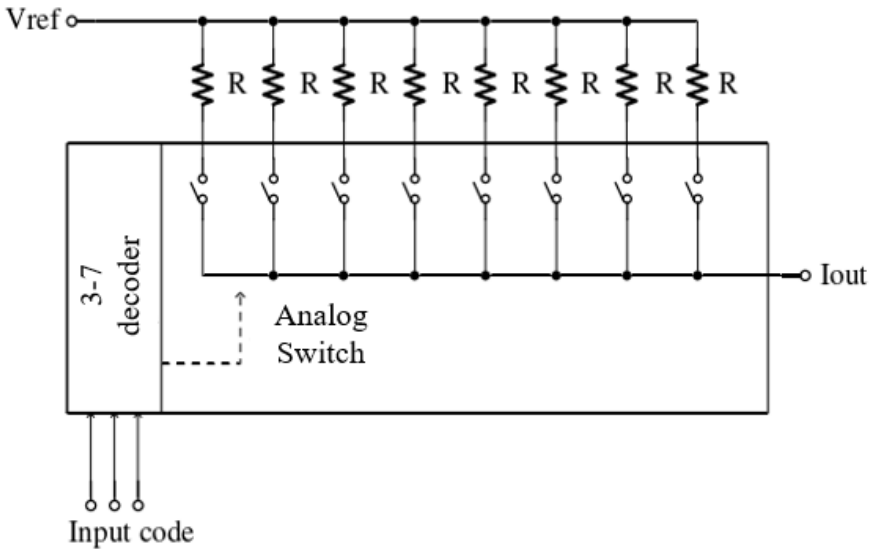


Fig. 2.4 Current output thermometer DAC.

If the resistor in Fig. 2.4 is changed to a current source, the Unary DAC will become the structure shown in Fig. 2.5. As mentioned at the beginning of this chapter, such a structure can also be called a current steering DAC.

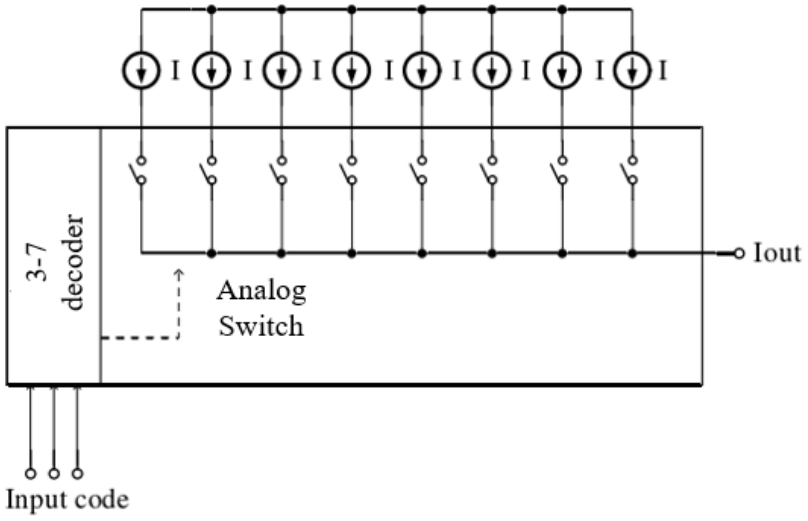


Fig. 2.5 Current steering thermometer DAC.

The current source in the current steering DAC generally consists of a group of current mirrors. Each output current is almost equal, and the error is much smaller than that of the resistance shunt thermometer DAC. This is the advantage of the current output thermometer DAC using the current mirror as the basic structure. Fig. 2.6 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.

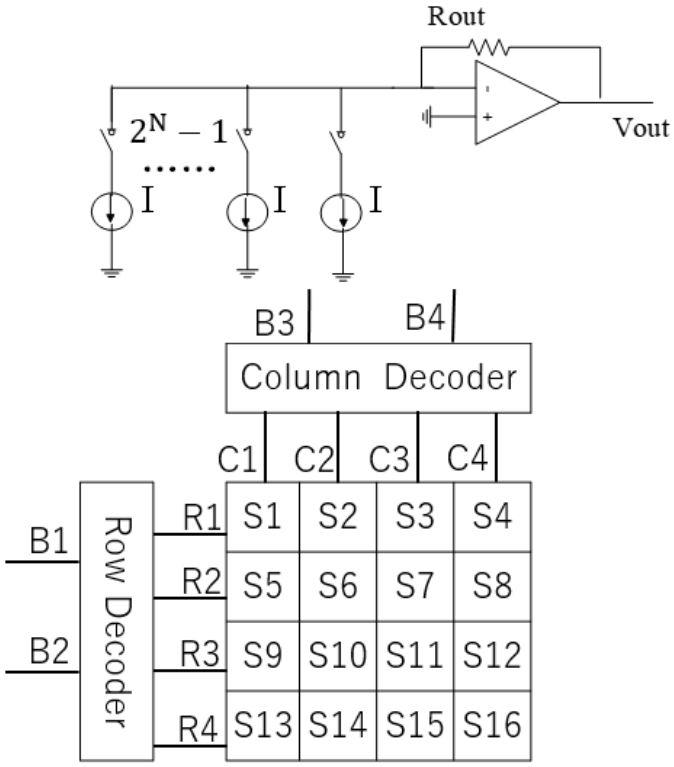


Fig. 2.6 Unary DAC circuit and regular layout of unit cell array. B4, B3, B2, B1 are binary inputs, while R1, R2, R3, R4, C1, C2, C3, C4 are their thermometer codes.

2.2.2 Binary DAC

The characteristic of the unary DAC is that the voltage difference of adjacent bits is the same, the output voltage with a large input code word value must not be less than the output voltage with a small input code word value, and the DAC itself has the advantage of monotonicity, which is why it is used. Obtained at the expense of a large number of resistors and a huge decoding circuit, the N-bit digital-to-analog converter requires 2^N resistors. The binary DAC structure overcomes this shortcoming. Like the unary DAC, the binary DAC structure also has two versions of voltage output mode and current output mode.

(1) Basic structure of binary DAC

Fig.2.7 is a DAC with binary structure in 4-bit voltage output mode. The entire structure only needs 4 resistors, and the size difference between adjacent resistors is doubled. This kind of DAC has a simple structure, but it does not have monotonicity, which makes it difficult to apply to the structure of high-precision DACs. The output resistance of the voltage mode binary DAC will change with the change of the input code. The multiple relationship between the maximum resistance and the minimum resistance in the structure is2^N, which increases the probability of resistance mismatch and makes the DAC more nonlinear. These are disadvantages of the basic structure of the binary DAC, or this is the price paid by the binary structure for reducing the number of resistors in the weighted network.

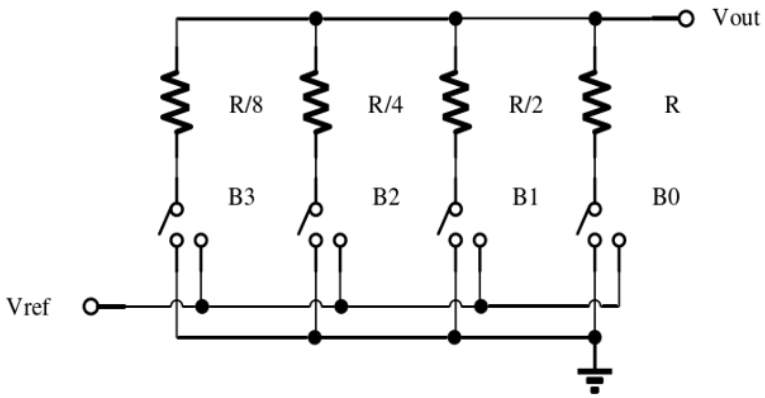
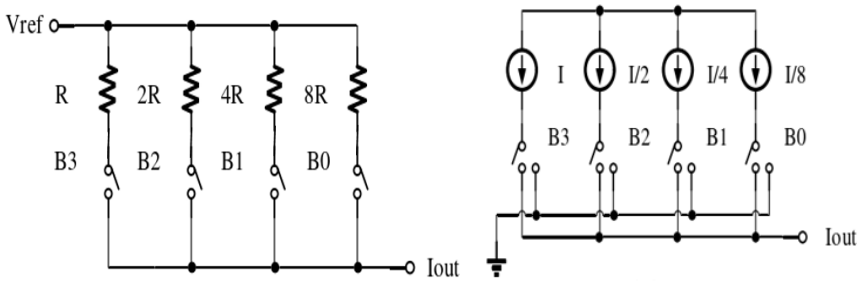


Fig. 2.7 4-bit voltage mode binary weighted DAC.

Fig. 2.8 is a block diagram of a binary DAC in current mode. This N-bit DAC is composed of N weighted current sources, and the current source can be composed of a MOS current mirror, or a resistor and a reference voltage source. The current-mode binary DAC also does not have monotonicity. If the current value of the MSB is low due to process errors, the MSB will be smaller than the sum of the currents of other bits, and the DAC will therefore exhibit nonlinearity.



(a) (b)

Fig. 2.8 (a) 4-bit resistor structure current mode binary weighted DAC. (b) 4-bit current steering structure current mode binary weighted DAC.

In the basic structure of binary DAC, whether it is voltage mode or current mode, the size of the highest bit resistance and the size of the lowest bit resistance are multiples of 2^N. This architecture is difficult to apply to the structure of high-precision DACs alone, but it can be used as a component of complex structure.

2.2.3 Segmented DAC

The basic structure and characteristics of DACs are introduced above, including unary DAC and binary DAC. The unary DAC has good static performance because of its inherent monotonicity, and the binary DAC has a simple structure but poor static performance. Table 2-1 lists the characteristics of the two basic structures of the DAC.

Table 2-1 Characteristics of the basic structure of DAC

Type	Structure	Output Mode	Advantage	Disadvantage
Unary DAC	String DAC	Voltage Output	Monotonic in structure	The logic circuit is complicated and the area is large, which is not suitable for high precision
	Thermometer DAC	Current Output	Monotonic in structure	The logic circuit is complex and not suitable for high precision
Binary DAC	Basic Type	Voltage Output	Simple structure, small area	Low precision, no monotonicity
		Current Output	Simple structure, small area	Low precision, no monotonicity
	R-2R Type	Voltage Output	Medium Accuracy, Constant Output Impedance	Input impedance changes, not monotonic
		Current Output	Medium Accuracy, Constant Output Impedance	Output impedance changes, not monotonic

We see from Table 2-1 that the advantage of the unary DAC is that it is inherently monotonic, which is determined by the structure of the unary DAC. The disadvantage is that as the number of bits increases, the required logical circuit area is large, which makes unary DAC not suitable for high-resolution applications. The binary DAC structure has two structures, namely the basic type and the R-2R type, neither of which has monotonicity, but the resistance of the R-2R type structure is only R and 2R, making it an important choice in the selection of binary DAC even better.

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

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

2.3 Performance Indicators

In order to describe the performance of DAC more comprehensively, some parameters and indicators will be introduced below. The indicators related to package design: differential nonlinearity (DNL), integral nonlinearity (INL), offset error and gain error are used to describe the static characteristics of the converter, while signal noise (SNR), total harmonic distortion (THD), spurious free dynamic range (SFDR), signal to noise plus

harmonic distortion ratio (SND) and significant bit (ENOB) are used to represent the dynamic characteristics of DAC.

2.3.1 Static State Characteristics of DAC

(1) Integral nonlinearity (INL)

INL is an index that indicates the cumulative error between the output transfer characteristic and the linear approximation straight line. (a) Use one of the two criteria shown in Fig. 2.9, namely the best fit line INL or (b) end point line INL [27].

The best fit line INL minimizes the absolute error value of INL by finding a reference line that best matches the transfer curve. The end line INL is the end point of the DAC transfer curve. The output voltage measured at (the minimum and maximum values of the input codes) is used to calculate the linearity of the DAC. In this example, the shape of the DAC transmission curve (measured value) of the two methods is the same. The endpoint method has only positive INL and no negative error, but the best slope and offset of the best fit line method are applied here, and the linear approximation line shifts upward. In this way, it can move part of the positive error to the negative side of the line and reduce the peak value of INL.

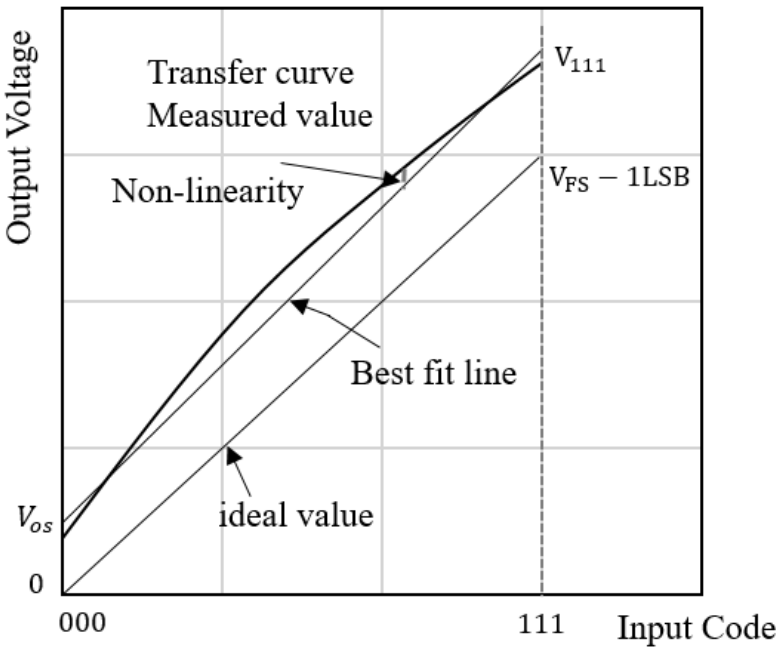


Fig. 2.9 DAC Integral Nonlinearity (INL) Reference.

Best fit line calculation

When calculating the INL, the best fit line is calculated according to the transfer curve. There are many methods, but the most commonly used method is the least square method, which can be obtained from the following Eqs. (2-2) - (2-5). In this calculation, in the case of 10-bit DAC, N must use all 2^{10}=1024 data sets, and the data of the code with bad results must not be sparse.

$$gain = \frac{N*K_4 - K_1*K_2}{N*K_3 - K_1^2} \tag{2-2}$$

$$offset = \frac{K_2}{N} - gain * \frac{K_1}{N} \tag{2-3}$$

$$K_1 = \sum_{i=0}^{N-1} i \, , K_2 = \sum_{i=0}^{N-1} S(i) \, , K_3 = \sum_{i=0}^{N-1} i^2 \, , K_4 = \sum_{i=0}^{N-1} i * S(i)$$

(2-4)

Where i is the input code and S (i) is the output voltage of the DAC. These equations can be easily implemented in the measurement program. The gain obtained by this formula is the voltage V_{LSB} of each quantization step.

The actual INL is composed of N data sets, and each value is calculated as the difference between the measured data normalized by the quantization step and the best fit line, as shown in Eq. (2-5), and its unit is LSB. Alternatively, it can be calculated using the sum of DNL obtained by Eq. (2-9) as in Eq. (2-6). It can also be calculated according to the end line or design criteria, but it is necessary to note that gain errors may occur in these methods.

$$\mathrm{INL}(i) = \frac{S(i) - (gain_{bestfit} * i + offset_{bestfit})}{V_{LSB}} [\mathrm{LSB}] \qquad (2\text{-}5)$$

$$\mathrm{INL}(i) = \sum_{j=0}^{i-1} DNL(j) \, [\mathrm{LSB}] \qquad (2\text{-}6)$$

Here, the gain error is generally represented by Eq. (2-7) as a ratio of deviation from the design specification of the gain. In addition, the offset error represents the deviation between the output of the DAC when the input code is 0 and the design value. In the case of binary encoding, the minimum output voltage V_{min}, and in the case of 2 complement encoding, the center output voltage represents the deviation from V_{mid}. It should be noted here that the 0 code encoded in 2 complement representation is actually a voltage higher than V_{mid} by 1/2 quantization step. The offset error is obtained by normalizing the difference from the design value by the quantization step size (V_{LSB}, $gain_{bestfit}$) obtained from the best fit line using Eq. (2-8). Gain and offset mismatch is one of the important parameters in quadrature modulation

DACs (Quadrature DAC), and it is desirable to achieve an offset error of 1 LSB or less.

$$gain_{error} = \frac{gain_{bestfit} - gain_{target}}{gain_{target}} 100[\%] \quad (2\text{-}7)$$

$$offset_{error} = \frac{(gain_{bestfit} * i_0 + offset_{bestfit}) - offset_{target}}{V_{LSB}} [\text{LSB}] \quad (2\text{-}8)$$

As described above, the INL measurements displays all the INL data corresponding to the ascending order code, or the INL minimum and maximum values for pass/fail judgment, as well as the gain error and offset error as measurement results.

(2) Differential Non-linearity (DNL)

DNL is an index representing the uniformity of the quantization step size obtained from the linear approximation straight line of the DAC output voltage difference between adjacent codes. Therefore, in DNL, the method of determining the quantization step size V_{LSB} affects the results. An important characteristic is whether the device exhibits non-monotonicity, such as when the error is less than -1 LSB as shown in Fig. 2.10 [28]. Normally, when the sample-to-sample output voltage differences of all adjacent codes are always increasing, that is, there are points where the output decreases when the input code increases, and points where the output increases when the input code decreases. When it does not exist, the DAC is said to have monotonicity.

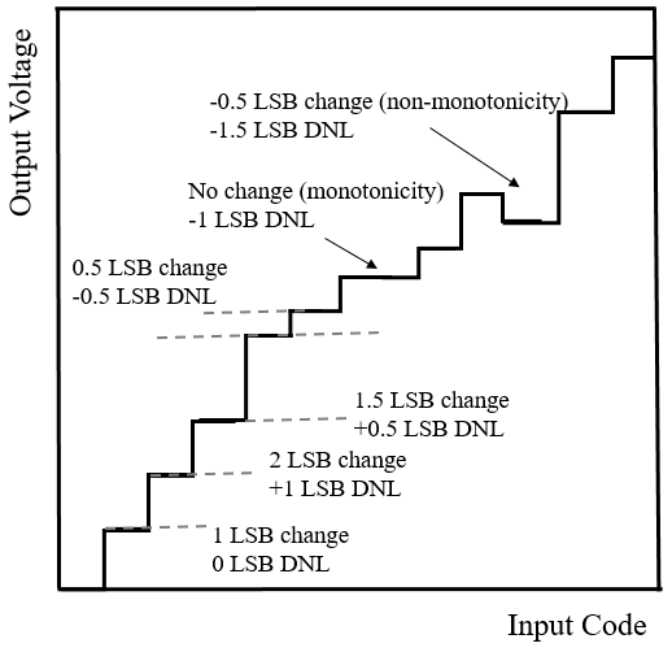


Fig. 2.10 DAC differential nonlinearity (DNL).

For the DNL calculation, N output voltage data sets for each code used for the INL calculation are used, and the obtained DNL is N-1 data sets. Each data is obtained by normalizing the difference between the actual output voltage difference and the quantization step size V_{LSB} by the quantization step, which is represented by the following Eq. (2-9). Here, V_{LSB} should use the value obtained from the best-fit line to remove the gain effect from the DNL index. V_{LSB} calculated by other methods have a negligible effect on DNL to some extent. The preferred method order for VLSB calculations is best fit line, end point line, design specification. On the other hand, the DNL can be expressed as the difference in INL between adjacent input codes as shown in Eq. (2-10).

In ADC measurement, INL can be easily obtained from DNL, while in DAC measurement, DNL is easier to obtain from INL.

$$\mathrm{DNL}(i) = \frac{S(i+1)-S(i)}{V_{LSB}} - 1 \ \ [\mathrm{LSB}] \tag{2-9}$$

$$\mathrm{DNL}(i) = \mathrm{INL}(i+1) - \mathrm{INL}(i) \quad [\mathrm{LSB}] \tag{2-10}$$

As described above, in the DNL measurement, all DNL data corresponding to two adjacent ascending codes or the minimum and maximum DNL values for pass/fail judgment are shown as measurement results.

2.3.2 Dynamic Characteristics of DAC

(1) Signal-to-Noise Ratio (SNR)

Signal to Noise Ratio (SNR) refers to the ratio of signal power Ps and noise power PN (excluding the energy of harmonic part) in analog output when a digital sine wave is input within a specific frequency range, expressed in dB, and can be expressed as:

$$\mathrm{SNR[dB]} = 10log_{10}(\frac{p_s}{p_n}) \tag{2-11}$$

The signal-to-noise ratio can also be expressed as the ratio of the amplitude of the full-scale sinusoidal analog output signal to the respective RMS value of the quantization noise. The signal-to-noise ratio (SNR) of an oversampling ADC can be written as follows:

$$\mathrm{SNR} = 10log_{10}\left(\frac{3}{2}2^{2N}OSR\right) = 10log_{10}(OSR) + 6.02N + 1.76 \quad (2\text{-}12)$$

(2) Total Harmonic Distortion (THD)

Total Harmonic Distortion (THD) refers to the ratio of the sum of the harmonic power of the DAC output signal to the fundamental power in a certain frequency band, which is expressed by the following formula:

$$\mathrm{THD} = 10\log_{10}\left(\frac{\sum_{k=2}^{m} p_k}{p_1}\right) \quad (2\text{-}13)$$

In the above formula, P_1 is the fundamental wave power of the signal, and P_k is the k-th harmonic power of the signal. Usually, only the first 10 harmonic components can be taken, while ignoring the harmonic components that have been attenuated to a very weak level more than 10 times (i.e. m=9 in (2-13)).

(3) Spurious-Free Dynamic Range (SFDR)

Spurious-Free Dynamic Range (SFDR) (Fig. 2.11) [29] is an important parameter for DACs for wideband communication. According to the obtained spectrum, as shown in Eq. (2-14), SFDR is expressed as the ratio of fundamental wave to maximum spurious. The signal power is Psig, and PHDM is the power of the harmonic that takes the maximum value.

$$\mathrm{SFDR} = 10\log_{10}\frac{\mathrm{p_{sig}}}{\mathrm{p_{HDM}}}[\mathrm{dB}] \quad (2\text{-}14)$$

The spectrum of DAC output signal will not only contain the expected signal frequency power, but also contain a lot of noise and harmonics. A

single-tone sine wave can be added at the input of the converter to measure SFDR. If the INL performance of the DAC is good enough, the SFDR value is usually higher than the signal-to-noise ratio.

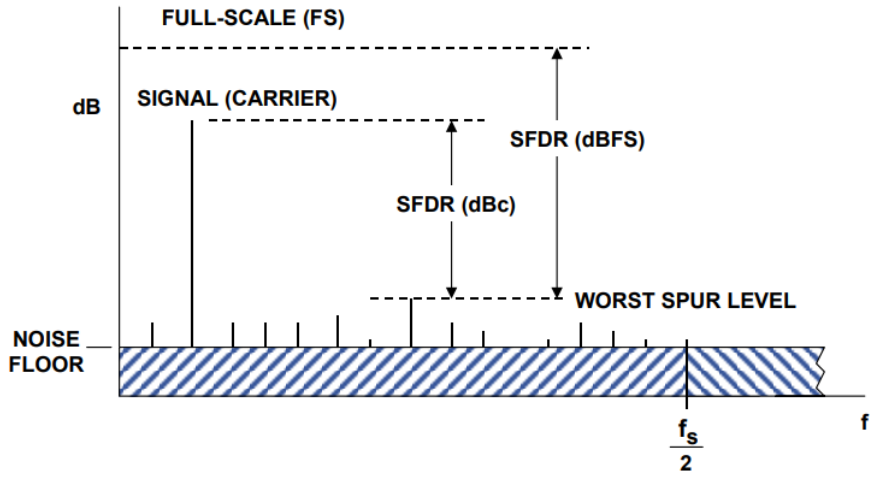


Fig. 2.11 Spurious-free dynamic range.

(4) Signal-to-Noise and Distortion Ratio (SNDR)

Signal-to-Noise and Distortion Ratio (SNDR) refers to the ratio of the signal power output by the DAC to the noise plus the sum of the harmonic signal power, in dB, can be expressed as:

$$\mathrm{SNDR} = 10 * \log \frac{P_1}{P_n + \sum_{k=2}^{\infty} P_k} \tag{2-15}$$

The difference between SNDR and SNR is that SNDR also contains other harmonic components except DC component.

(5) Effective Number of Bits (ENOB)

The ideal is determined by the number of bits of resolution. In reality, due to nonlinear factors, its dynamic characteristics are reduced compared

with the ideal. Therefore, it is more suitable to measure its dynamic characteristics than to measure it in practice. It becomes ideal when the harmonic distortion is reduced to, so in order to measure the deviation between the harmonic distortion and the ideal, the degree of linearity reflected is introduced. The defined expression is:

$$\text{ENOB} = \frac{SNDR[dB]-1.76}{6.02} \tag{2-16}$$

2 .4 Circuit Element Characteristics

2.4.1 Variations in Circuit Element Characteristics

There are systematic variations among MOSFET, resistor and capacitor characteristics on an integrated circuit, due to their placements, 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 could be nonlinear. The causes of these variations are as follows [3,10-14]:

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 Fig. 2.12, Fig. 2.13 and Fig. 2.14. (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. 2.12)

$$\varepsilon_l(x,y) = g_l * \cos\theta * x + g_l * \sin\theta * y \qquad (2\text{-}17)$$

θ: angle of inclination g_l:magnitude of the slope

2) Quadratic error (Fig. 2.13)

$$\varepsilon_q(x,y) = g_q * (x^2 + y^2) - a_0 \qquad (2\text{-}18)$$

g_q:variable value a_0:position

3) Linear and quadratic joint errors (Fig. 2.14)

$$\varepsilon_j(\mathrm{x},\mathrm{y}) = \varepsilon_l(\mathrm{x},\mathrm{y}) + \varepsilon_q(\mathrm{x},\mathrm{y}) \qquad (2\text{-}19)$$

In general, the coefficients of x^2 and y^2 may be different. We choose Eq. (2-17).The influence of the systematic variation on the DAC linearity can be mitigated by technique to the target unit cell selection algorithm or layout.

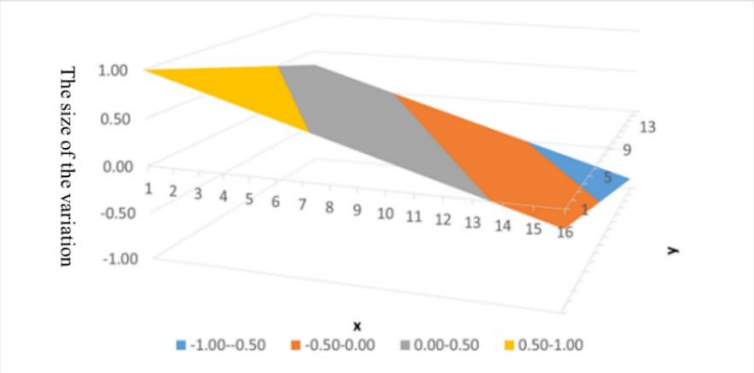


Fig. 2.12 Linear gradient error.

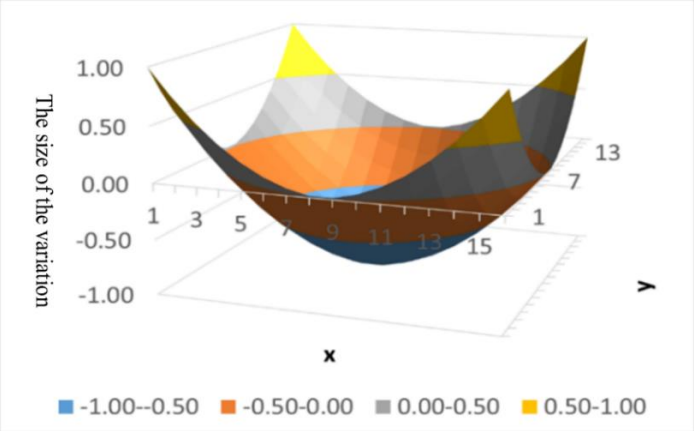


Fig. 2.13 Quadratic gradient error.

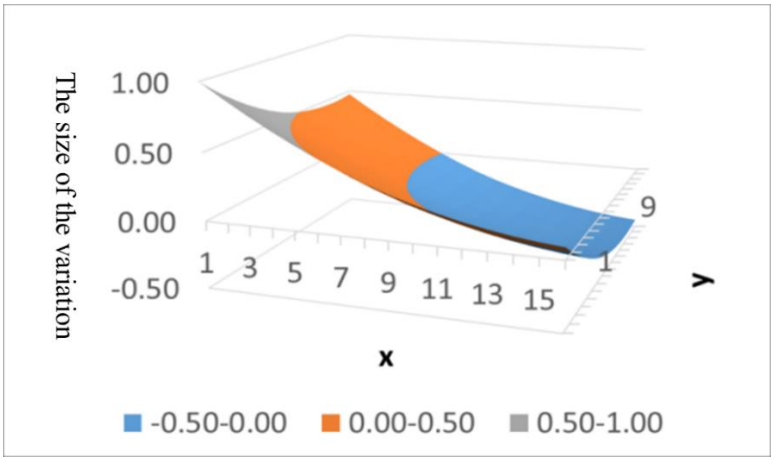


Fig. 2.14 Linear and quadratic joint gradient errors.

2.4.2 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.2.15) [4]. Ideal drain current I_d in saturation region is given by:

$$I_{d,1} = \frac{\beta}{2}(V_{gs} - V_{th1})^2 \qquad (2\text{-}20)$$

Also drain current mismatch is given by;

$$\frac{\Delta I_{d1}}{I_{d1}} = \frac{2}{V_{gs} - V_{th1}} \frac{Av_{th}t_{ox}}{\sqrt{WL}} \qquad (2\text{-}21)$$

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 ($\sqrt{WL}$).

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 (Fig.2.16, Fig.2.17).

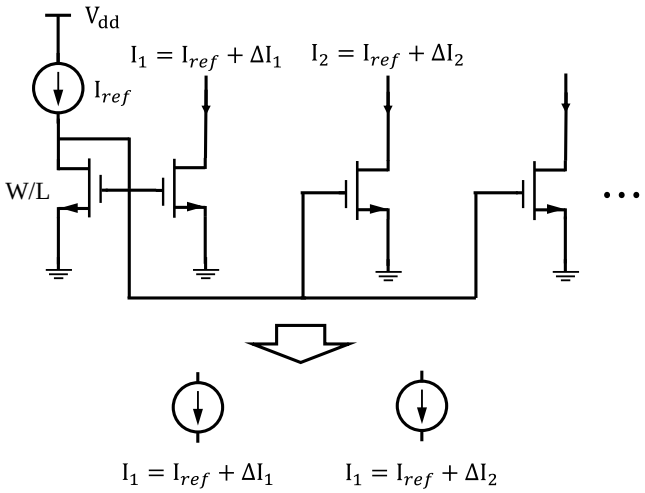


Fig. 2.15 Mismatches among current sources.

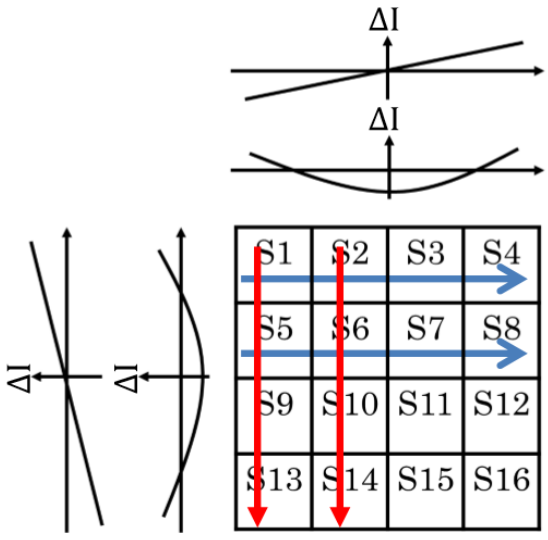


Fig. 2.16 Regular layout of unit cells for a unary DAC and their linear/quadratic gradient errors.

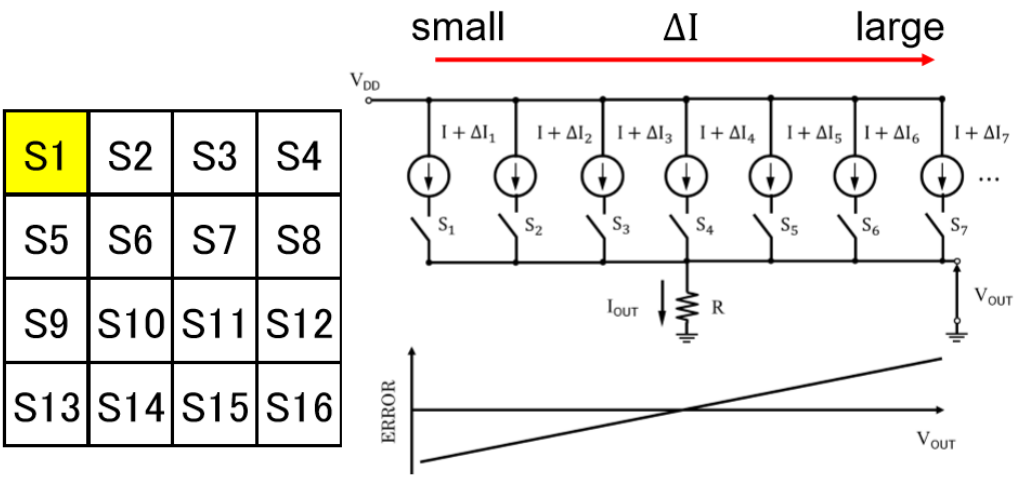


Fig. 2.17 Regular layout of unit cells for a unary DAC and its nonlinearity due to their linear gradient errors

2.5 Summary

In this chapter, the working principle of the DAC is introduced in detail. The input digital code controls the on-off of the analog switch, and the digital code is added according to the weight of each bit to convert the digital quantity into the corresponding analog quantity.

The basic structure of the DAC can be divided into a unary DAC and a binary DAC according to the relationship between adjacent bits. The advantage of the unary DAC is that it has monotonicity, and the disadvantage is that the logic circuit is more complicated. The logic circuit of the binary DAC is simple, but the static parameters are not ideal. Segmented DAC

refers to a DAC with two or more structures, which can have the advantages of multiple single-structure DACs at the same time.

The index parameters of the DAC include static parameters and dynamic parameters. We discuss the influence of components in a circuit on IC wafer.

References

[19] Shen Qisong. High-order oversampling delta-sigma DAC design. Chengdu: University of Electronic Science and Technology of China (2011).

[20] Rako P. Using Analog Switches to Realize Signal Multiplexing. Electronic Design Technology, 8(9): 72-78 (2008).

[21] Zhu Xinyue. Design of DAC chip anti-irradiation multi-voltage mode amplifier [D]. Xi'an, Xidian University (2018).

[22] Fu Zhibo. Research on high-speed and high-precision DAC design. Beijing: North China University of Technology (2013).

[23] Kim K D, Park C B, Lee S W, et al. 35.3: A 10-bit Linear R-string DAC Architecture for Mobile Full-HD AMOLED Driver ICs //SID Symposium Digest of Technical Papers. Oxford, UK: Blackwell Publishing Ltd, 44(1): 469-472 (2013).

[24] Saponara S, Baldetti T, Fanucci L, et al. High-level modeling of resistor string based digital-to-analog converters. Analog Integrated Circuits & Signal Processing, 2011, 66(3):407-416.

[25] J.B.Simoes，J.Landeckand, C.M.B.A.Correia,“NonlinearityofaData 一 Acquisition System with Interieaving / MultiPlexing”, IEEE Trans. Instrumentation and Measurement, vol.46, no.6, pp.1274-1279 (Dec.1997).

[26] Backer R, Li H W, Boyce D E. CMOS Circuit Design, Layout, and Simulation. Beijing: Machinery Industry Press, pp.136-141 (2003)

[27] Scott Wayne, "高分解能 D/A コンバータを最大限に活用する方法,” Application Note, AN-313,Analog Devices

[28] “差動出力,電流モードディジタル-アナログコンバータ(DAC)のリニアリティの測定,” Application Note, AN4159, Maxim Integrated Products.

[29] W. Kester, “The Data Conversion Handbook", 2005, ISBN 0-7506-7841-0. (Also published as Analog-Digital Conversion, Analog Devices, Inc. 2004, ISBN 0-916550-27-3.)

Chapter 3

Random Selection Algorithms of Unit Cells in Unary DAC

3.1 Magic Square

A Magic Square has a property that the sums of each row/ column/diagonal element 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 which can reduce the systematic variation effects to improve the DAC linearity (Fig. 3.1).

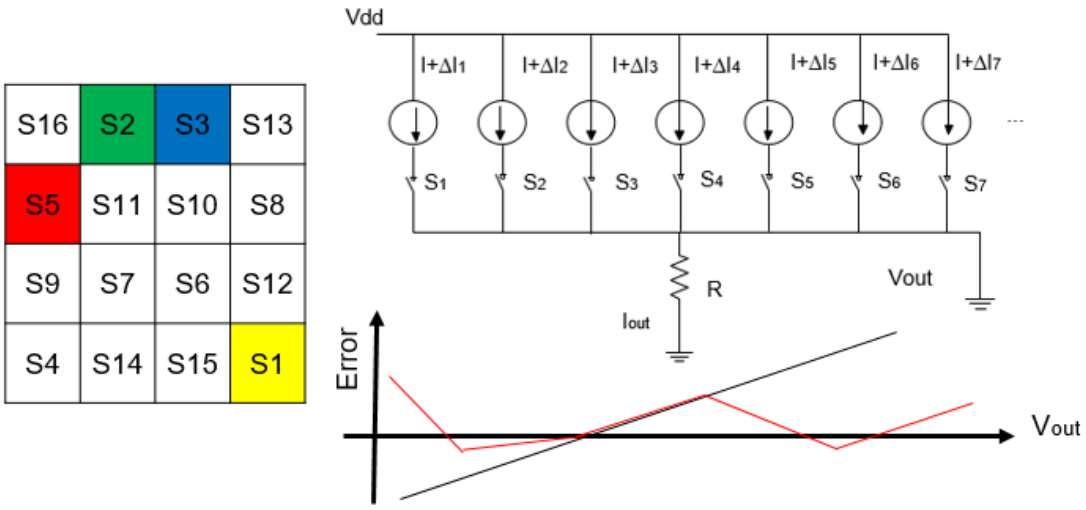


Fig. 3.1 Layout technique of unit cells for a unary DAC and its linearity improvement by cancelling their linear gradient errors.

3.1.1 Features of Magic Square

The Magic Square with the *n* x *n* matrix, which is arranged in a grid pattern, is a series of natural numbers starting from 1 to *n* x n while the numbers on each row, column or diagonal elements have equal sum [5-7]. 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:

$$S = \frac{n^2(n^2+1)}{2} \tag{3-1}$$

In the Magic Square shown in Fig. 3.2, it can be confirmed that the sum of the elements of each row, column and diagonal components are all equal.

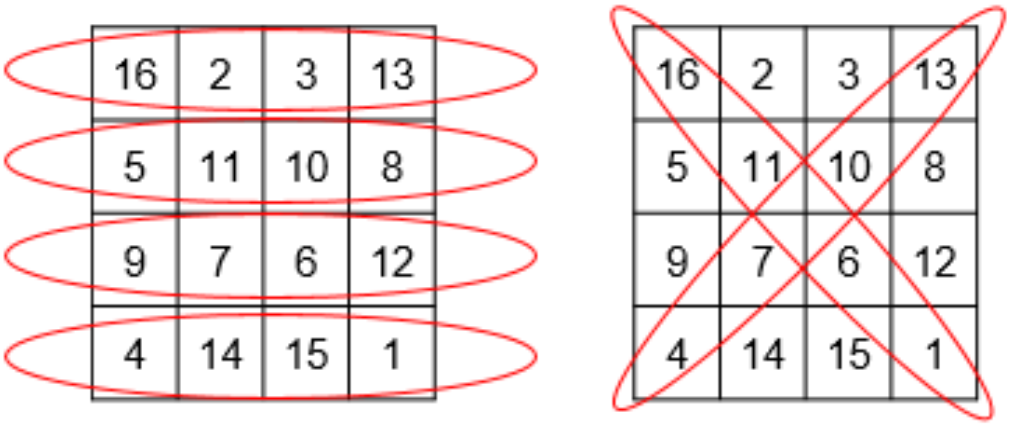


Fig. 3.2 Equivalent constant sum characteristics.

3.1.2. Algorithm Using Concentric Magic Square

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. 3.3, and the

DAC linearity improvement with the layout based on the Magic Square was confirmed by simulation.

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. 3.3 16x16 Magic Square.

3.2 Latin Square

3.2.1 Characteristics of Latin Square

Latin Squares have n rows and n columns of n different symbols, and are arranged in such a way that each symbol appears only once in each row and each column (Fig. 3.4 (a)) [7-9]. 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. 3.4 (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

Fig. 3.4 (a) 4x4 Latin Square.

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

Fig. 3.4 (b) 4x4 Complete Latin Square.

3.3 Euler Knight Tour

3.3.1 Features of Euler Knight Tour

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.3.5 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.3.6 shows an example. The Euler Knight Tour has both properties of Magic Square and Knight Tour as shown in Fig. 3.7.

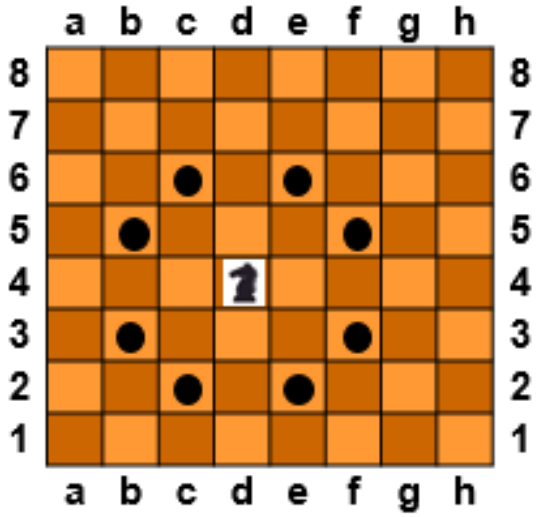


Fig. 3.5 Places where a Knight piece can move starting from around the center of an 8x8 chess board.

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. 3.6 Euler Knight Tour algorithm in an 8x8 matrix.

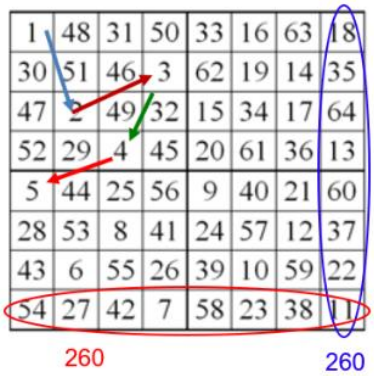


Fig. 3.7 Euler Knight Tour algorithm in an 8x8 matrix.

3.4 Simulation Result

The unary DAC with regular layout allows variations of the linear and quadratic gradients to degrade its linearity. Fig. 3.8 shows the case of 8-bit (16 × 16). We compared this regular layout, the Magic Square layout (Fig. 3.9), the Latin Square layout (Fig.3.10) and the Euler Knight Tour layout (Fig .3.11).

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. 13.12) and INL for the case of quadratic gradient error (Figs. 3.13 (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. 3.14-3.17); 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.3.12 and the results in the quadratic gradient variation are shown in Figs.3.13 (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. 3.8 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. 3.9 Magic Square layout of 2D array current cells.

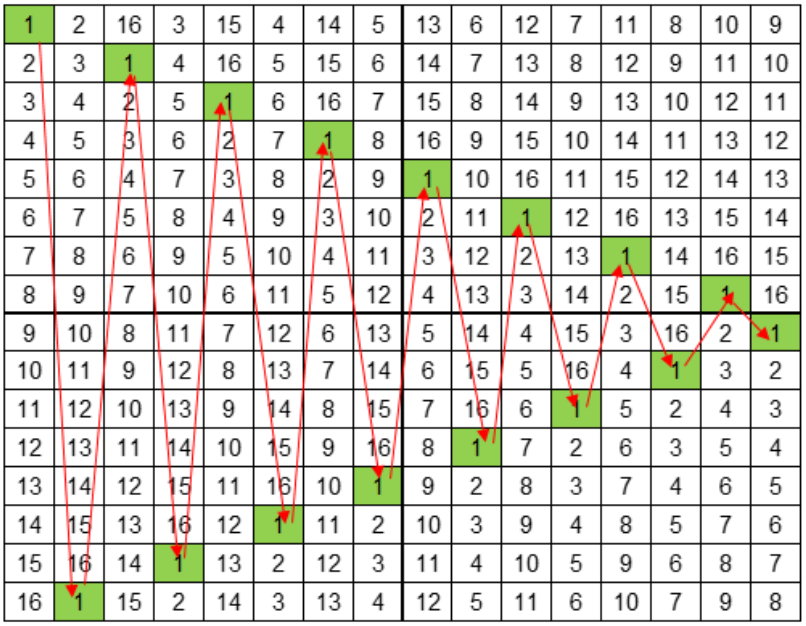


Fig. 3.10 Complete Latin Square layout of 2D array of current cells.

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

Fig. 3.11 Euler Knight Tour layout of 2D array current cells.

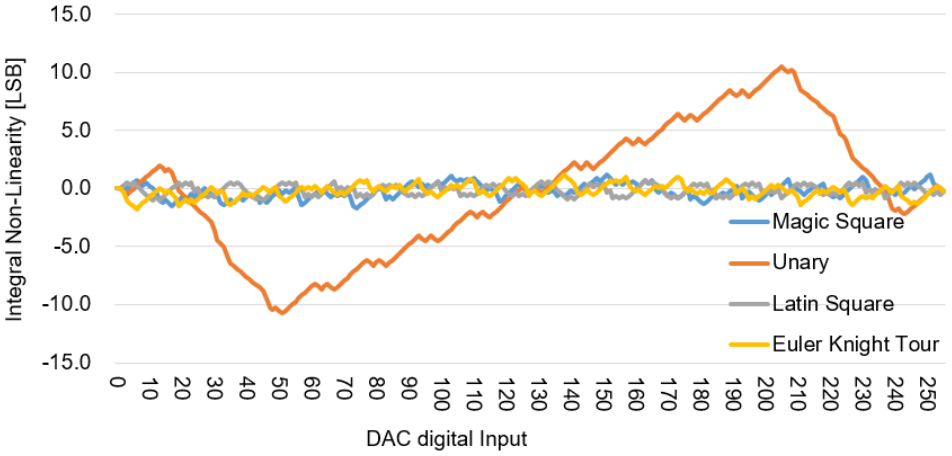
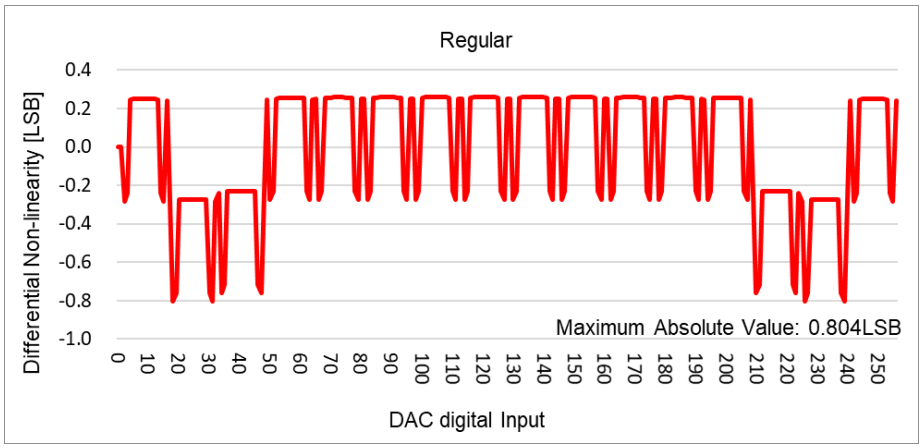
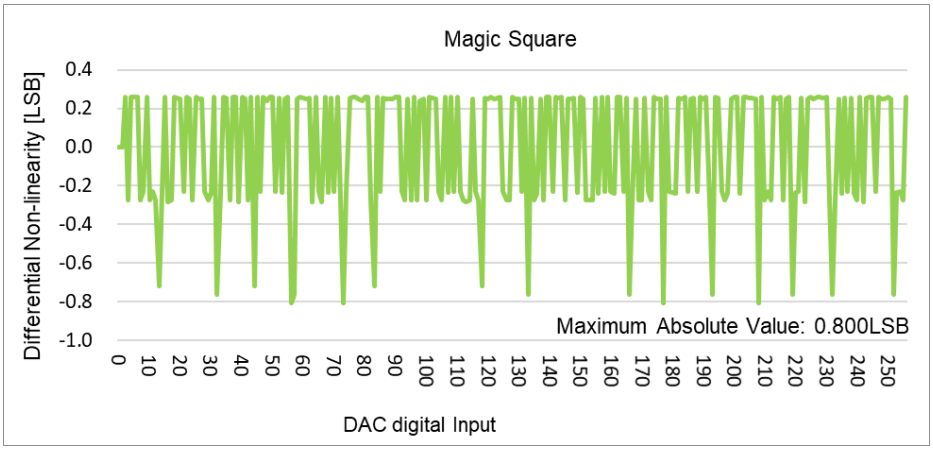


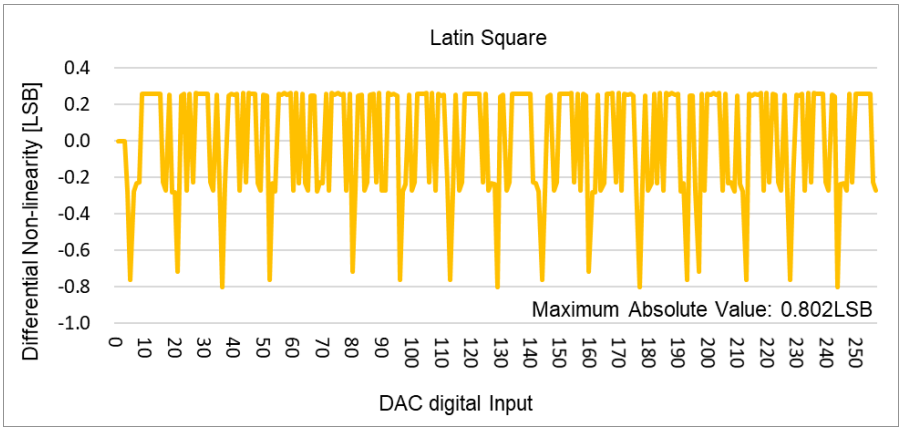
Fig. 3.12 Simulated INL in case of linear gradient.



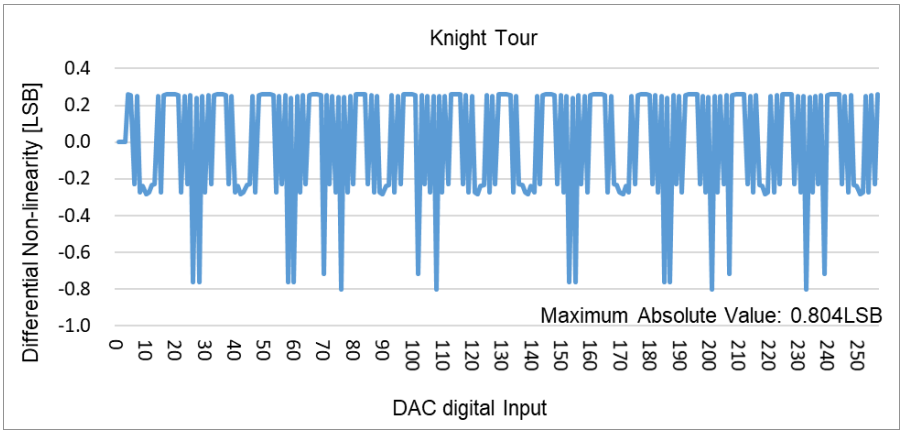
(a) Regular layout case.



(b) Magic Square case.



(c) Latin Square case.



(d) Knight Tour case.

Fig. 3.13 Simulated INLs for quadratic gradient error.

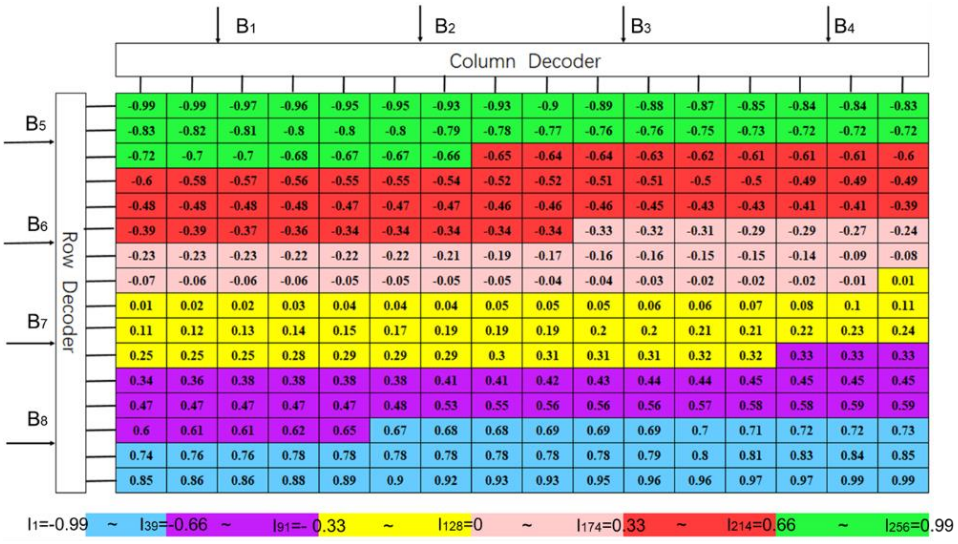


Fig. 3.14 Simulation condition is distribution of variation. (Top to bottom, left to right, small to large.)

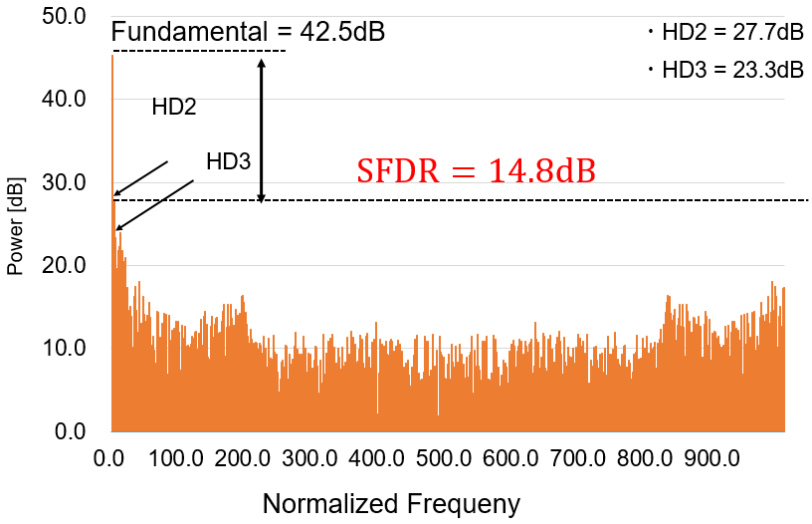


Fig. 3.15 SFDR for the Regular layout.

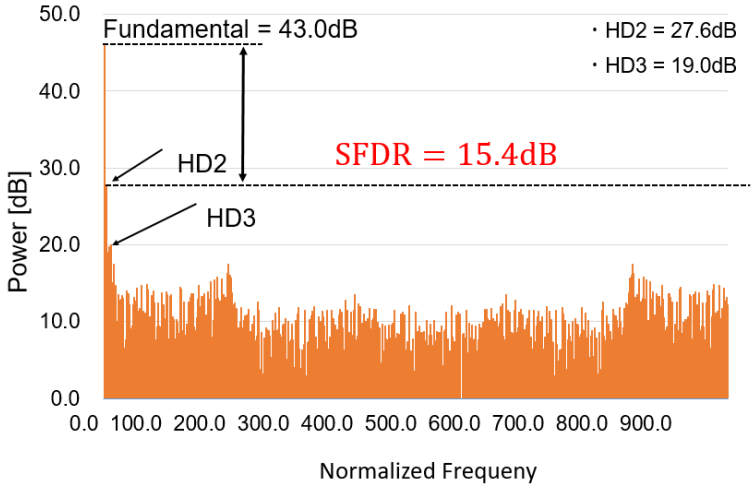


Fig. 3.16 SFDR for the Magic Square layout.

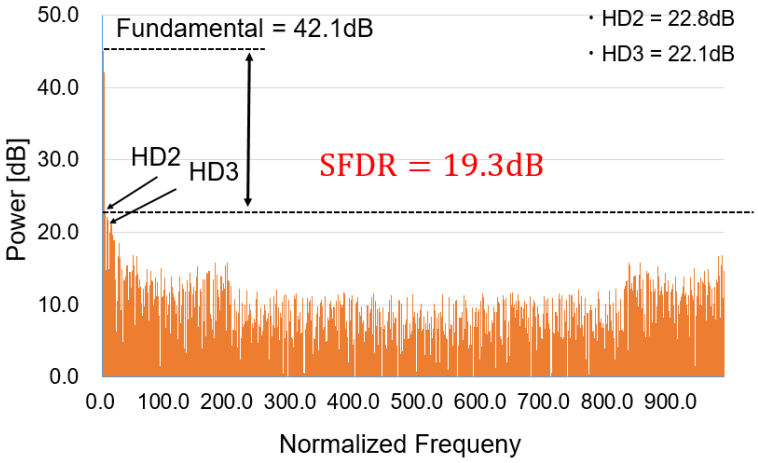


Fig. 3.17 Simulated SFDR for complete Latin Square case.

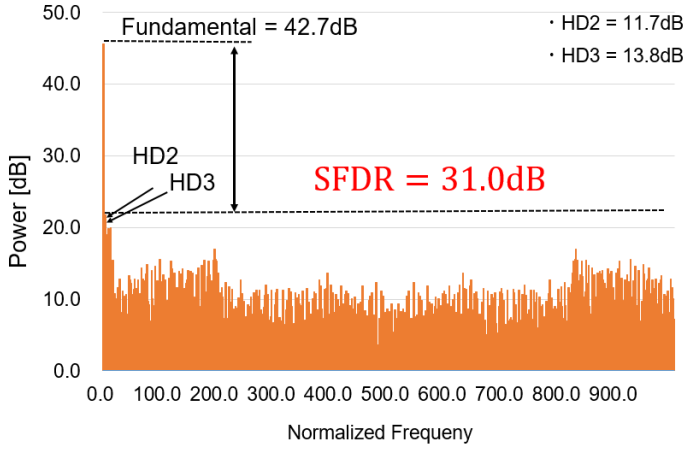


Fig. 3.18 Simulated SFDR for Euler Knight Tour layout case.

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.3.20-3.21, and the results were obtained in Figs.3.22-3.23. The abscissa is the value range of SFDR, and the ordinate is the number of this value range. It can be seen from the results that the results of the Latin Square

and Magic Square in the quadratic distribution are better than the regular distribution. It can be improved by about 5dB to 10dB (Fig. 2.22). But the alignment of the linear and quadratic distributions, although improved, is not so obvious (Fig. 2.23). We see that when the variation obeys the linear distribution (Fig.3.19), Latin Square is the best, followed by Magic Square, Euler Knight Tour, and Regular. In the quadratic distribution case (Fig.3.20), 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.3.21), the Latin Square and Euler Knight Tour are the best followed by Magic Square and then Regular.

Notice in Figs3.15–3.18 that “Fundamental" is NOT normalized to 0dB, but is direct FFT calculation result because it can change depending on randomization algorithms; the fundamental component power can be spread in frequency domain due to randomization.

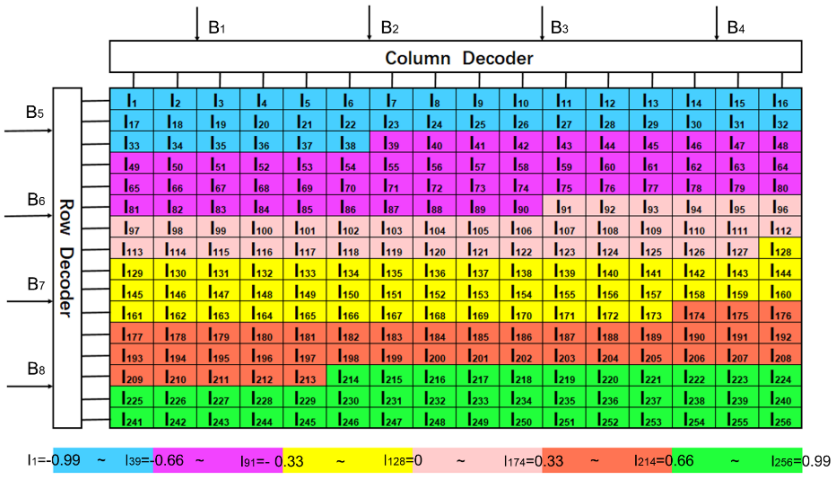


Fig. 3.19 Simulation condition is linear distribution of variation. (Top to bottom, left to right, small to large.)

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

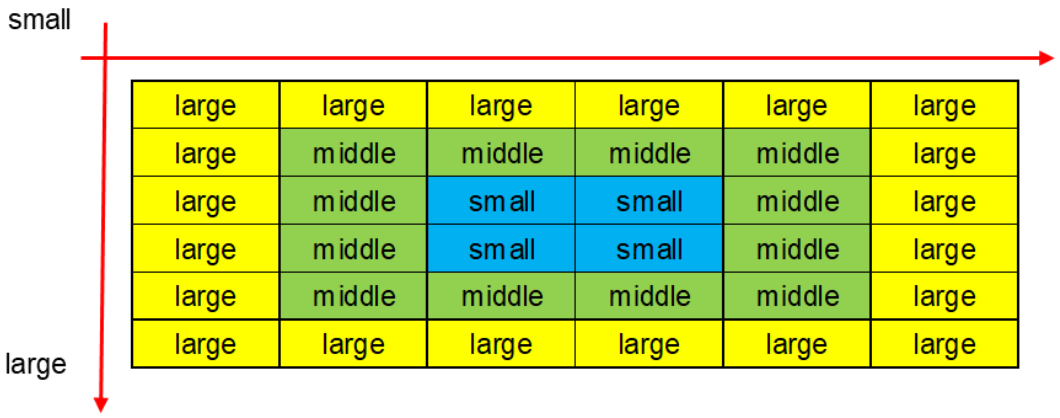


Fig. 3.21 Simulation condition is linear + quadratic distribution of variation. (From the center to the periphery from small to large and top to bottom, left to right, small to large)

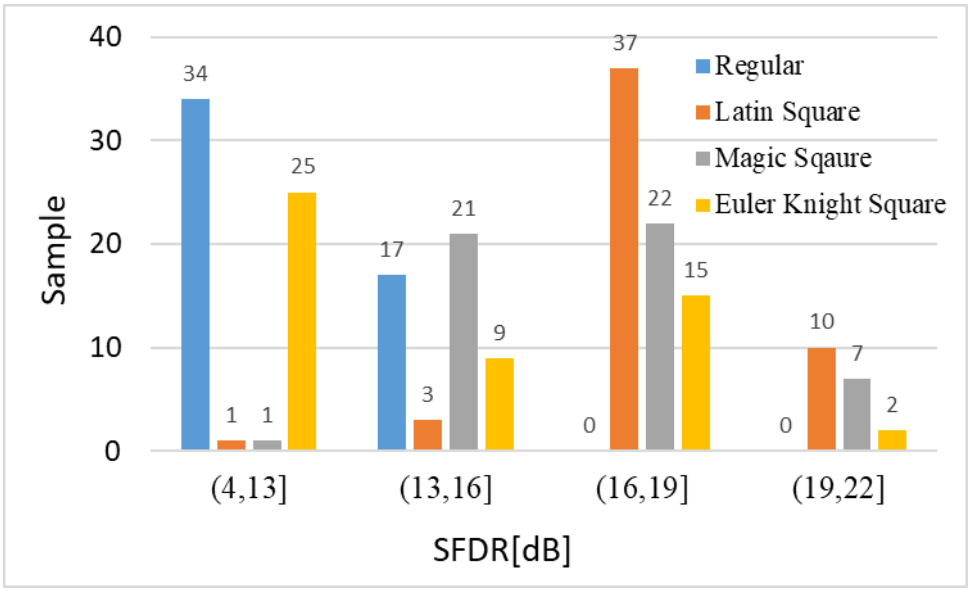


Fig. 3.22 Simulated SFDR result with quadratic variation using 50-Monte Carlo analysis.

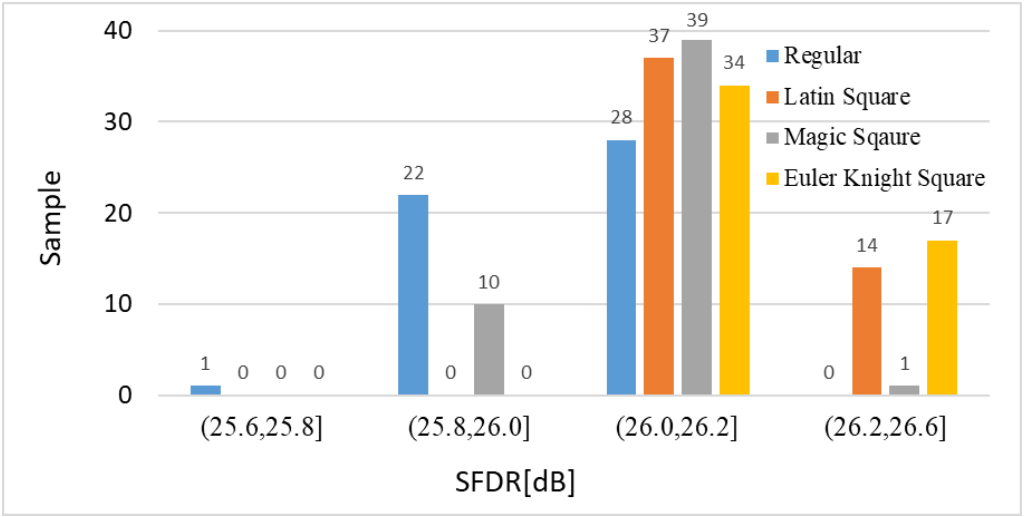


Fig. 3.23 Simulated SFDR result with linear + quadratic variation using 50-Monte Carlo analysis.

3.6 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. 3.24 shows the floor plan of the 2D unit cell array for the 6-bit (8x8) case while Fig. 3.25 shows the 2D regular layout case [9]; Fig.3.26 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.

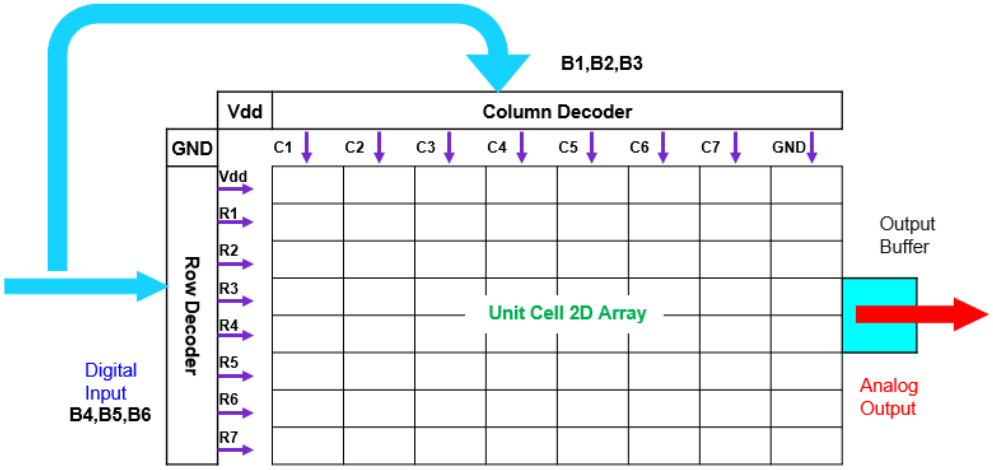


Fig. 3.24 6-bit unary DAC floor plan.

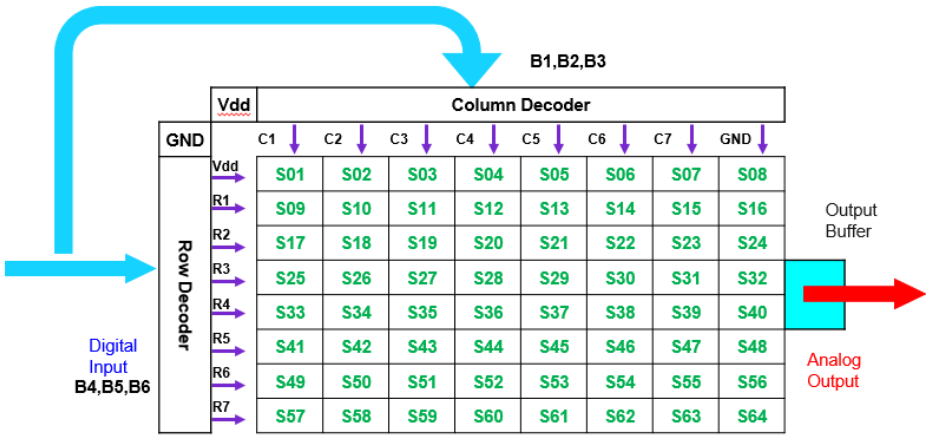


Fig. 3.25 Unary DAC floor plan for 2D regular unit cell array.

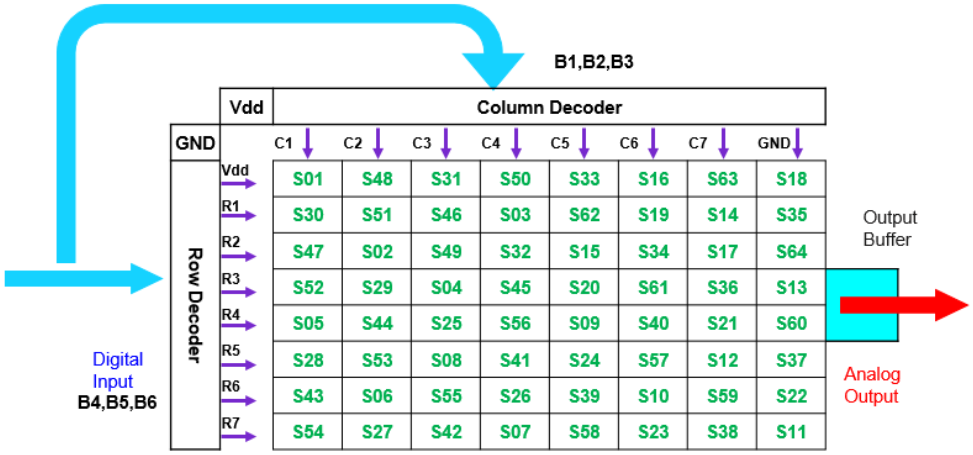


Fig. 3.26 Unary DAC floor plan for 2D unit cell array based on Euler Knight Tour.

Fig.3.27-3.30 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.

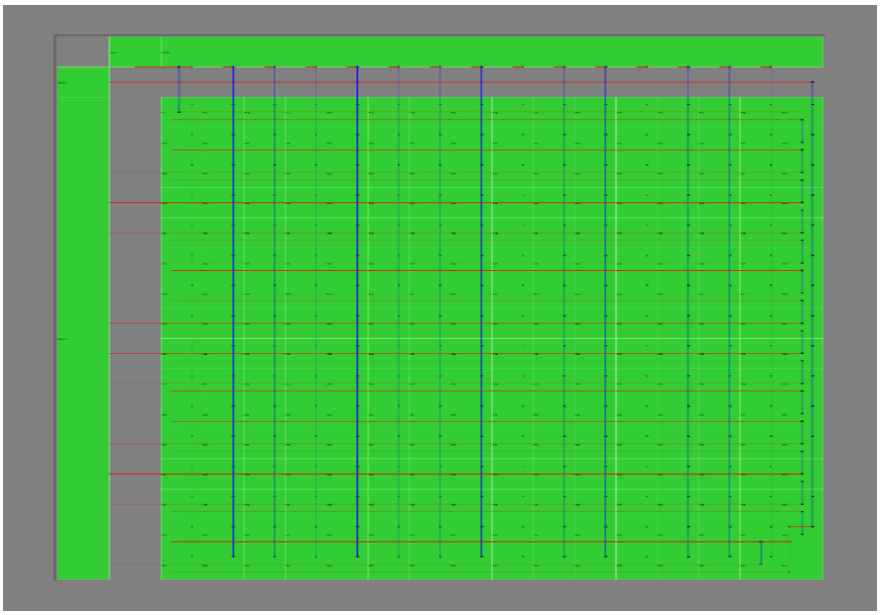


Fig. 3.27 8-bit (16x16) unit cell 2D layout and routing based on Regular square.

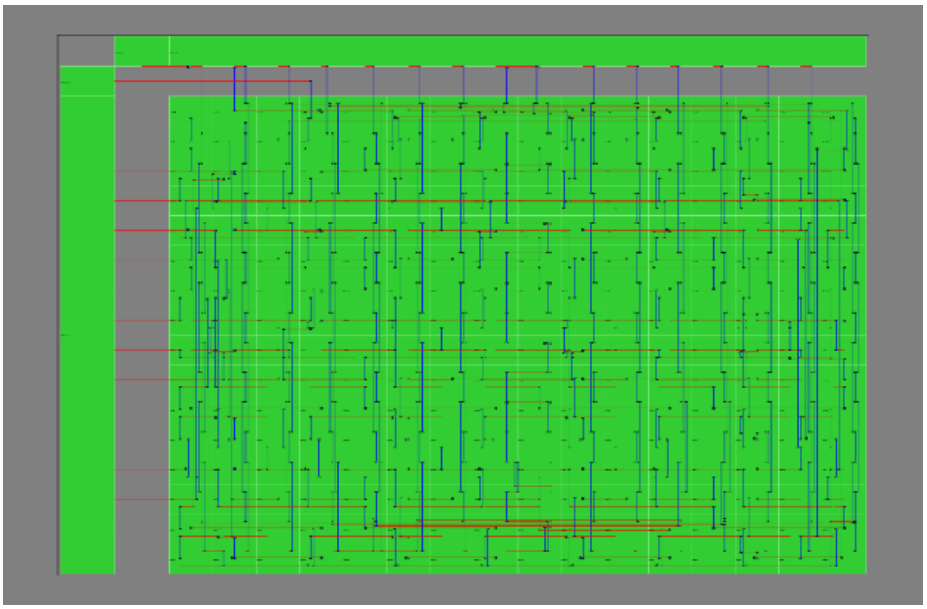


Fig. 3.28 8-bit (16x16) unit cell 2D layout and routing based on Magic Square.

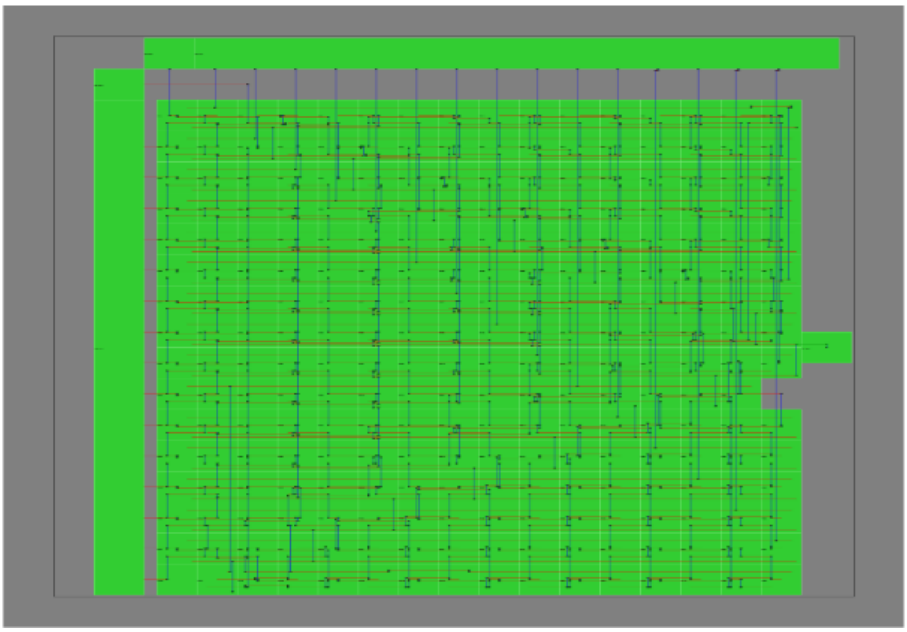


Fig. 3.29 8-bit (16x16) unit cell 2D layout and routing based on complete Latin Square. The irregular part of the most right is the output buffer.

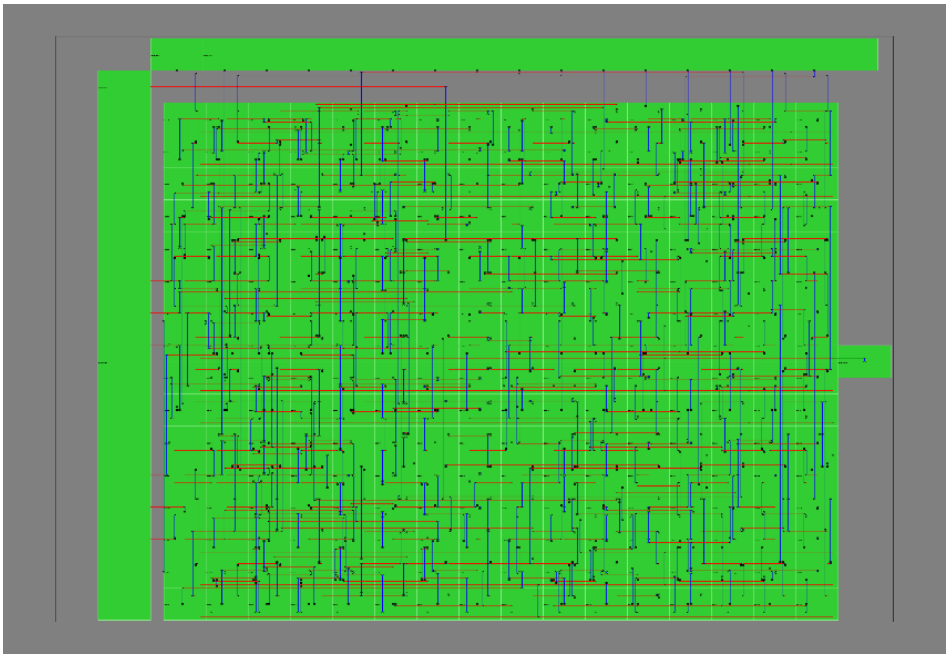


Fig. 3.30 8-bit (16x16) unit cell 2D layout and routing based on Euler Knight Tour square. The irregular part of the most right is the output buffer.

3.7 Summary

3.7.1 Conclusion

In this dissertation, 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.

3.7.2 Items for Future Study

The conclusions obtained through computer simulation and the layout of the circuit board were all before. Later, it can be designed and verified through FPGA. If the result of the verification is satisfactory, the physical design and verification can be carried out.

Chapter 4

Mismatch Scrambling Algorithms of Unit Cells in Unary DAC

4.1 Introduction

Digital-to-Analog Converter (DAC) for communication applications requires good Spurious Free Dynamic Range (SFDR) performance [27, 28]. The SFDR of the unary DAC is degraded due to static characteristics mismatches among its unit cells as well as its dynamic characteristics.

In this dissertation, we focus on the care for the static characteristics mismatches using the Dynamic Element Matching (DEM) for the unary DAC. We consider here the signal bandwidth of the DAC is from DC to the Nyquist rate (half of the sampling frequency) or higher frequency, so that the DEM technique only needs to spread the spectrum of the spurious tones caused by the mismatches in the entire signal band uniformly without need for mismatch shaping; this is called as mismatch scrambling [29].

However, direct implementation circuit of such mismatch scrambling for the high-resolution unary DAC becomes complicated. Then, in [30] a mismatch scrambling technique utilizing the features of 2D regular (square or rectangular) layout and routing for the unit cells was investigated, utilizing row and column decoders features [31].

Here, we investigate a more hardware efficient method for a higher resolution DAC, by considering a virtual 3D case, considering X-,Y- and Z-decoders.

Notice that most of DEM algorithms are for low resolution DACs and for mismatch shaping [29, 32-56]; to our knowledge, there is no research for small hardware mismatch scrambling for high resolution DACs except for [30].

4.2 Configuration and Problems of Unary DAC

DAC architectures may be classified into binary and unary types as well as their combination (segment type), as shown in Fig.4.1 [30] [31]. 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 a unit array of voltage, charge or current. DA conversion is performed by summing these unit cell outputs. Fig.2.6 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 (especially for DNL) on the output signal than the binary type even if there are mismatches among elements. The glitch is small and monotonicity characteristics is guaranteed in principle. However, its disadvantages are relatively 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.2.6)

become a problem. Thus, the cell selection technique for alleviating this influence is necessary.

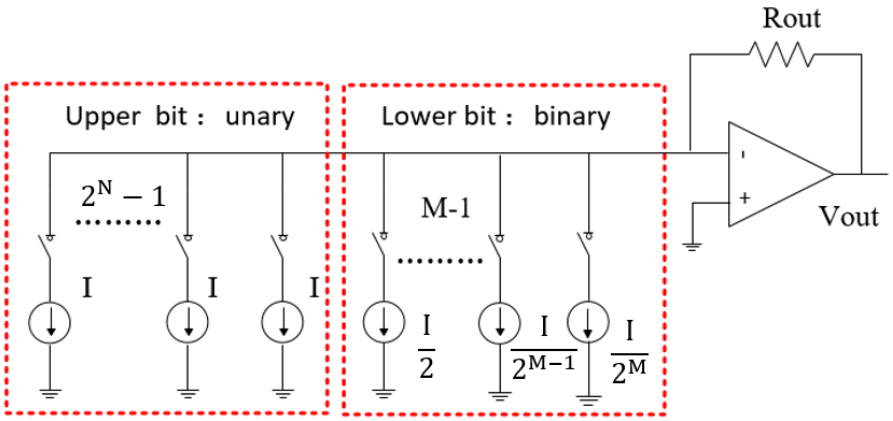


Fig. 4.1 Segmented DAC with binary and unary structures.

Many DACs consist of combination of unary and binary types; Unary type is used for the higher bits, while the binary type with a small number of elements is for lower bits. Then a high performance DAC with appropriate circuit scale and power can be realized. Hence, we consider only the unary type because the unary type handles higher bits, which influences overall linearity of the segmented DAC [31].

We consider here a 2-dimensional (2D) matrix layout of unit cells as shown in Fig.2.6 [34]. The column decoder converts the upper binary bits to the thermometer codes, while the row decoder does the lower binary bits to the thermometer ones. Then we have 2D regular layout and routing of unit cells.

Fig.4.1 shows a 6-bit unary DAC with unit current sources. Its binary digital inputs (B_6, B_5, B_4, B_3, B_2, B_1) are converted into the thermometer codes (S_{63}, S_{62}, ..., S_2, S_1) through the decoder. Here we assume the following:

$$I = I_1 = I_2 = \cdots = I_{63}.$$

Then the k-th current switch for S_k=1 turns on. For example, if the digital input 000111, then S_k=1 (k=1,..,7), S_m=0 (m=8,..,63), and the analog output is V_{OUT}= 7 R I.

Next, we consider mismatches among current sources in Fig.4.1. I_n is the unit current source corresponding to S_n. Ideally all values of I_n (n=1, 2,.., 63) are identical, but in reality, these can be different [32]:

$$I_n = I + \Delta I_n$$

$$\text{Here } I = \left(\frac{1}{63}\right)(I_1 + \ I_2 + \cdots + I_{63})$$

These mismatches ΔI_n cause the overall DAC nonlinearity and generate spurious components of the DAC output in frequency domain and thus degrade the SFDR.

Then the mismatch scrambling method is considered to spread the spurious spectrum components caused by the mismatches and improve the selection at each sampling time in a pseudo random manner.

For example, consider the case that the digital input is the DC value of 00101. Then a mismatch scrambling algorithm may work as follows:

At time k, S_5=S_4=S_3=S_2=S_1=1

And the analog output

$$V_{OUT}(k) = R(5I \ + \Delta I_5 + \Delta I_4 + \Delta I_3 + \Delta I_2 + \Delta I_1).$$

At time k+1, S_{28}=S_{21}=S_{17}=S_{11}=S_7=1 and

$$V_{OUT}(k+1) = R(5I \ + \Delta I_{28} + \Delta I_{21} + \Delta I_{17} + \Delta I_{11} + \Delta I_7).$$

At time k+2, S_{60}=S_{51}=S_{37}=S_{23}=S_8=1 and

$V_{OUT}(k+2) = R(5I \ +\Delta I_{60}+\Delta I_{51}+\Delta I_{37}+\Delta I_{23}+\Delta I_8)$

Here the current mismatch effects are dynamically changed.

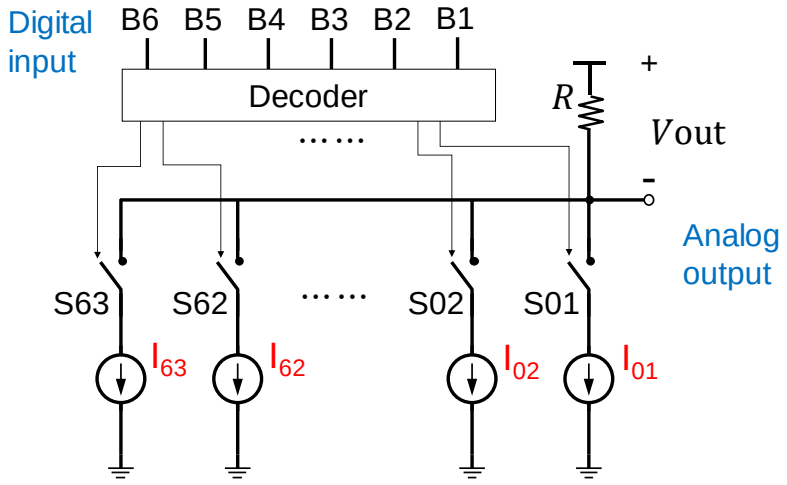


Fig. 4.2 6-bit unary DAC configuration.

4.3 Dynamic Element Matching Technique

The DEM is a technique to change the unit cell selection order and spread the DAC output spurious in frequency domain; it can improve the SFDR [31] [36] . However, the circuit to implement conventional DEM increases exponentially as the number of unit cells increases. Here we investigate its simple implementation utilizing the structure of the 2D matrix layout of the unit cells.

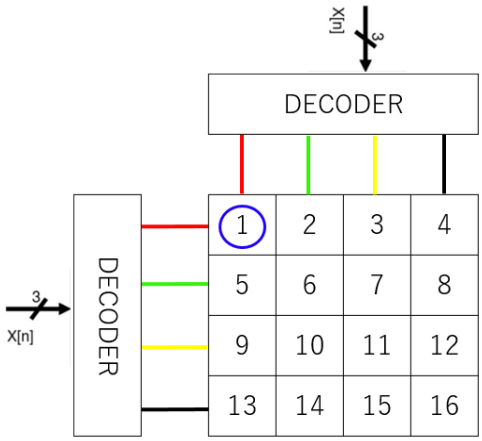


Fig. 4.3 Unary DAC with 2D layout of unit cells (without DEM).

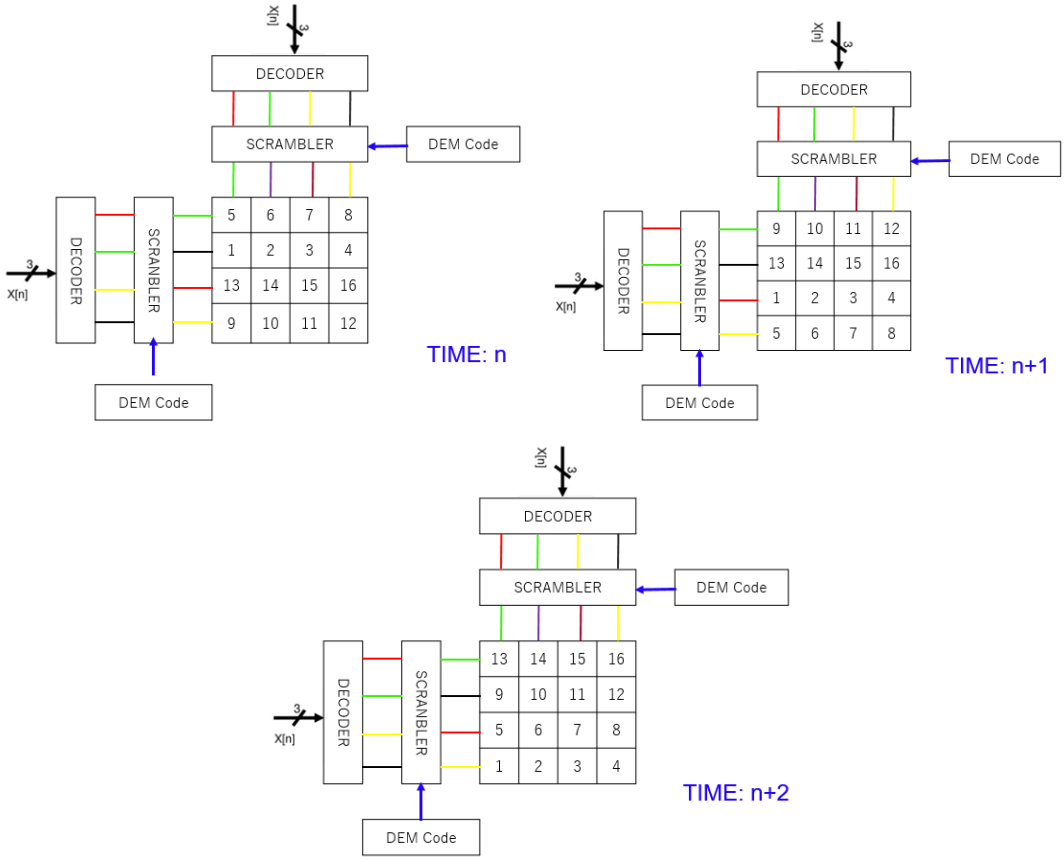


Fig. 4.4 Unary DAC with 2D layout of unit cells (with DEM).

Fig.4.4 shows 2D matrix layout of unit cells for a 4-bit unary DAC. Fig. shows the proposed 2D matrix layout of unit cells and circuit for a 4-bit unary DAC, where scrambler circuits are added after the row and column decoders. The scrambler circuit can be implemented by a multiplexer array with the addition of a linear feedback shift register circuit for controlling its selection signals in a pseudo random manner; these can realize a DEM algorithm.

Fig.4.5 shows the change for the unit cell selection order and Fig.4.6 shows the corresponding simulation results.

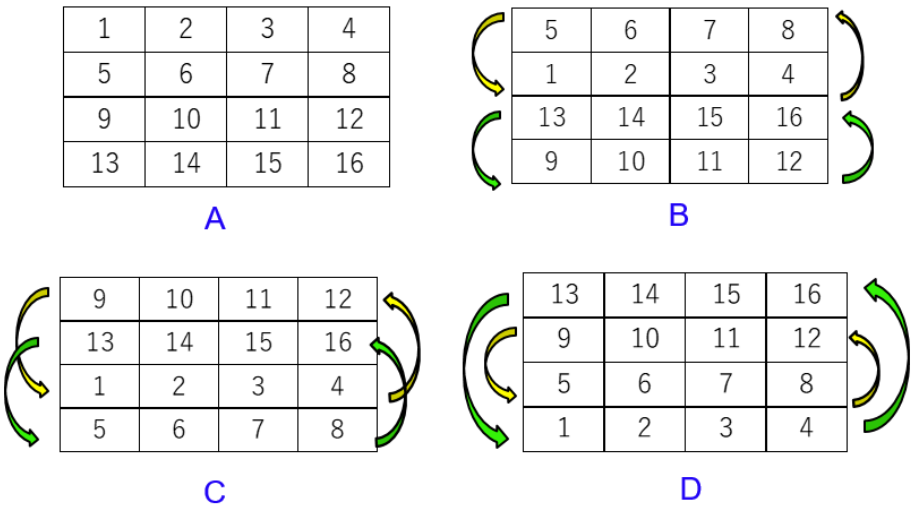


Fig. 4.5 Unary DAC cell selection algorithm used for simulation.

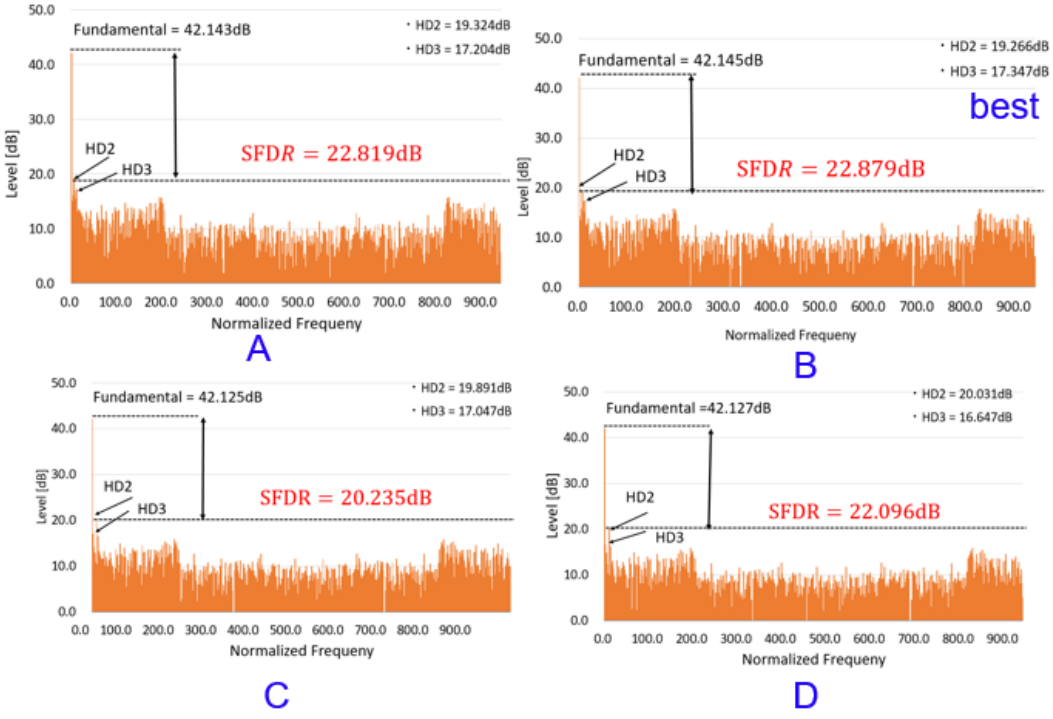


Fig. 4.6 Simulated DAC output power spectrum without DEM.

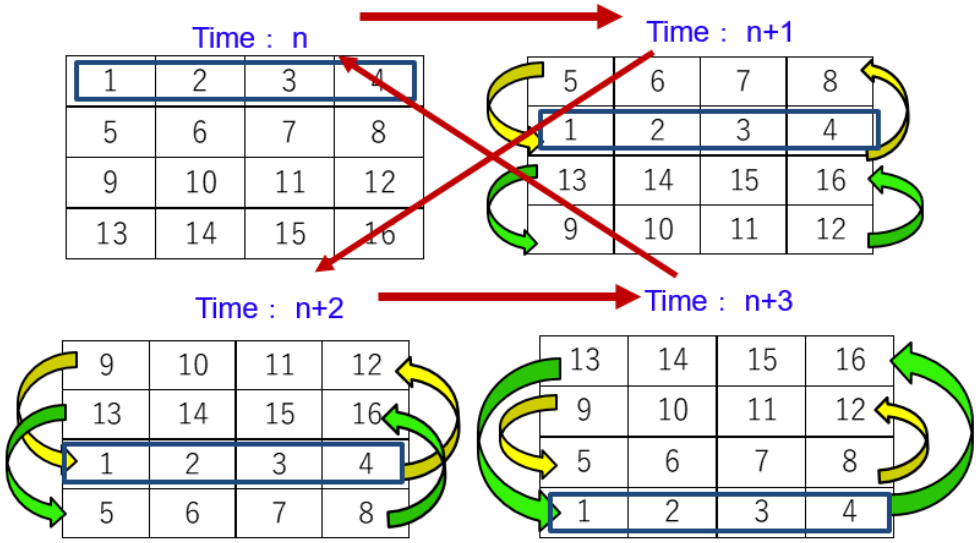


Fig. 4.7 Unary DAC DEM algorithm used for simulation.

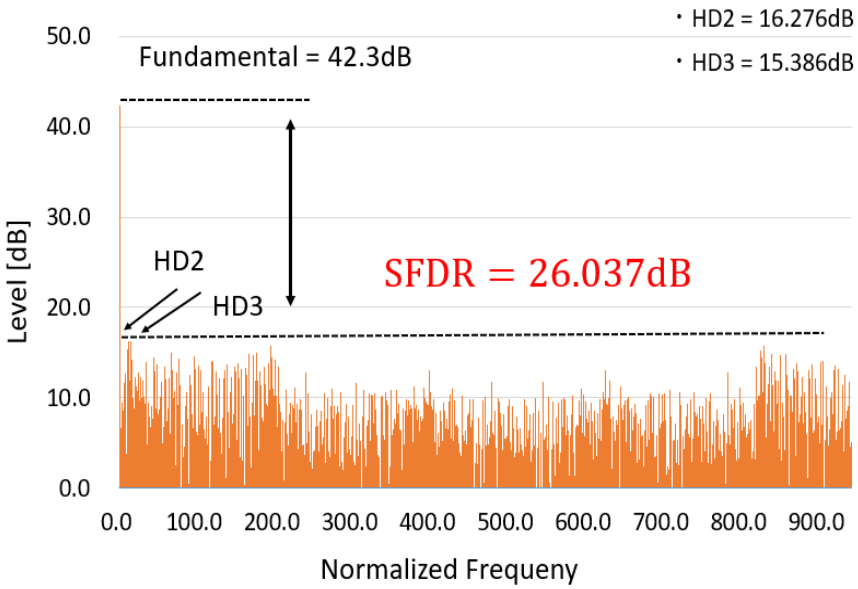


Fig. 4.8 Simulated DAC output power spectrum with DEM.

Table 4-1 Simulated SFDR comparison with and without DEM.

Device	SFDR (dB) DEM OFF	SFDR (dB) DEM ON	Improverment
A	22.819	26.037	3.218
B	22.879		3.158
C	20.235		5.802
D	22.096		3.941

We have simulated a 4-bit unary DAC with the proposed DEM (only column decoder output shuffling) as shown in Fig.4.7, and its output power spectrum is shown in Fig.4.8. Comparison among SFDR values are shown in Table.4-1.

4.3 Mismatch Scrambling Technique for 2D Regular Layout

The circuit to implement conventional mismatch scrambling increases exponentially as the number of unit cells increases if it is directly implemented. Hence its simple implementation utilizing the features of the 2D matrix layout and routing of the unit cells is discussed in [30]. Our interpretation for 2D regular layout mismatch scrambling method is as follows:

We consider here a 6-bit unary DAC with digital binary inputs of B_6, B_5, B_4, B_3, B_2, B_1, and their thermometer codes are S_{64}, S_{63}, S_{62}, …, S_3, S_2, S_1. Also consider the 2D square layout and routing of the unit cells as shown in Fig.4.9 There, each component is given as follows:

i) Row-decoder: binary inputs B_6, B_5, B_4,

thermometer outputs R_7, R_6, R_5, R_4, R_3, R_2, R_1.

$$R_7=B_6B_5B_4,\ R_6=B_6B_5,\ R_5=B_6(B_5+B_4),\ R_4=B_6,$$

$$R_3=B_6+B_5B_4,\ R_2=B_6+B_5,\ R_1=B_6+B_5+B_4.$$

ii) Column-decoder: binary inputs B_3, B_2, B_1,

thermometer outputs C_7, C_6, C_5, C_4, C_3, C_2, C_1.

$$C_7=B_3B_2B_1,\ C_6=B_3B_2,\ C_5=B_3(B_2+B_1),\ C_4=B_3,$$

$$C_3=B_3+B_2B_1,\ C_2=B_3+B_2,\ C_1=B_3+B_2+B_1.$$

See Table 4-1. Also we define $R_0=C_0=1$, $R_8=C_8=0$

We use a local decoder for each unit cell with the following logical expression (Table 4-3):

$$S_n = R_{p+1} + R_p C_q$$

S_n is the switch for *n*-th current source. Here,

For $1 \leq n \leq 8$, $p = 0$.

For $9 \leq n \leq 16$, $p = 1$.

For $17 \leq n \leq 24$, $p = 2$.

For $25 \leq n \leq 32$, $p = 3$.

For $33 \leq n \leq 40$, $p = 4$.

For $41 \leq n \leq 48$, $p = 5$.

For $49 \leq n \leq 56$, $p = 6$.

For $57 \leq n \leq 64$, $p = 7$.

For $1 \leq \mathrm{mod}_8\ (n) \leq 7$, $q = \mathrm{mod}_8\ (n)$.

For $\mathrm{mod}_8\ (n) = 0$, $q = 8$.

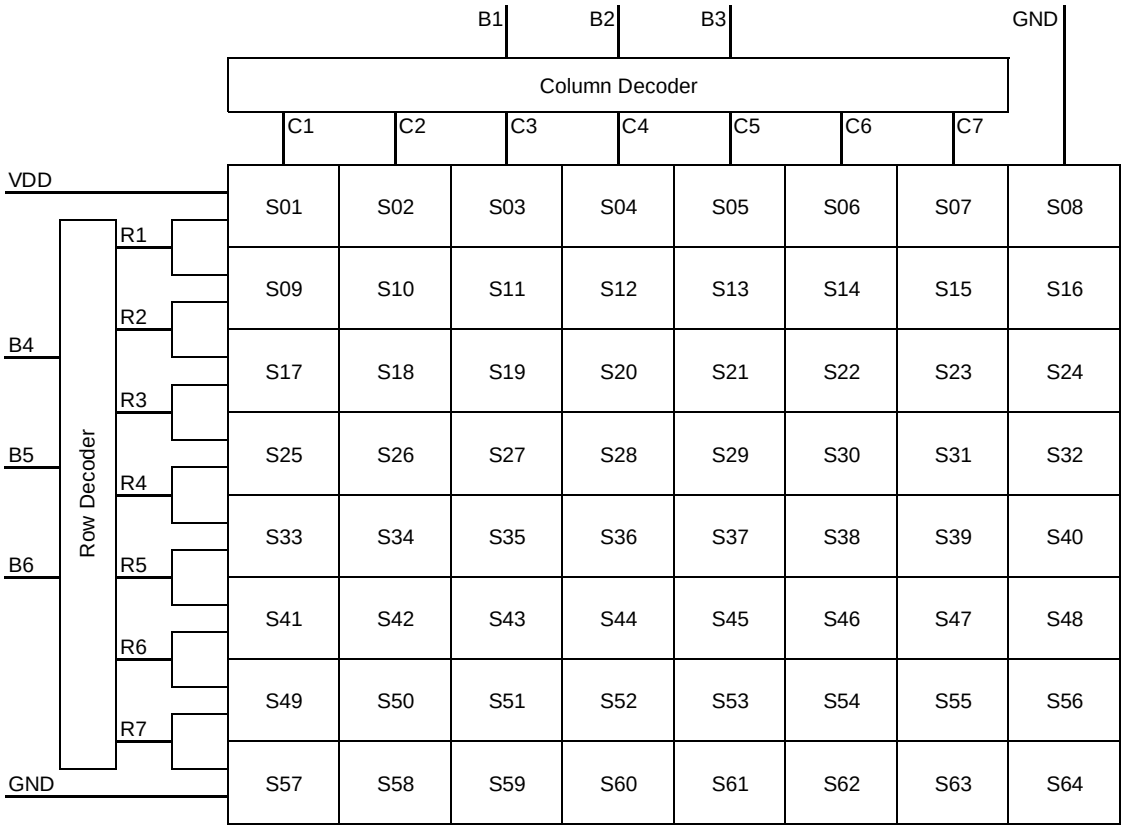


Fig. 4.9 2D unit cell layout of 6-bit unary DAC.

Table 4-2 Truth tables of binary-to-thermometer decoders for upper 3-bit (left) and lower 3-bit (right) of a 6-bit unary DAC.

B6	B5	B4	R7	R6	R5	R4	R3	R2	R1
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	1
0	1	1	0	0	0	0	1	1	1
1	0	0	0	0	0	1	1	1	1
1	0	1	0	0	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

B3	B2	B1	C7	C6	C5	C4	C3	C2	C1
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	1
0	1	1	0	0	0	0	1	1	1
1	0	0	0	0	0	1	1	1	1
1	0	1	0	0	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Table 4-3 Logical expressions of unit cell local decoders for a 6-bit unary DAC with 2D layout in Fig.4.3.

S01 = R1 + 1*C1	S17 = R3 + R2*C1	S33 = R5 + R4*C1	S49 = R7 + R6*C1
S02 = R1 + 1*C2	S18 = R3 + R2*C2	S34 = R5 + R4*C2	S50 = R7 + R6*C2
S03 = R1 + 1*C3	S19 = R3 + R2*C3	S35 = R5 + R4*C3	S51 = R7 + R6*C3
S04 = R1 + 1*C4	S20 = R3 + R2*C4	S36 = R5 + R4*C4	S52 = R7 + R6*C4
S05 = R1 + 1*C5	S21 = R3 + R2*C5	S37 = R5 + R4*C5	S53 = R7 + R6*C5
S06 = R1 + 1*C6	S22 = R3 + R2*C6	S38 = R5 + R4*C6	S54 = R7 + R6*C6
S07 = R1 + 1*C7	S23 = R3 + R2*C7	S39 = R5 + R4*C7	S55 = R7 + R6*C7
S08 = R1 + 1*0	S24 = R3 + R2*0	S40 = R5 + R4*0	S56 = R7 + R6*0
S09 = R2 + R1*C1	S25 = R4 + R3*C1	S41 = R6 + R5*C1	S57 = 0 + R7*C1
S10 = R2 + R1*C2	S26 = R4 + R3*C2	S42 = R6 + R5*C2	S58 = 0 + R7*C2
S11 = R2 + R1*C3	S27 = R4 + R3*C3	S43 = R6 + R5*C3	S59 = 0 + R7*C3
S12 = R2 + R1*C4	S28 = R4 + R3*C4	S44 = R6 + R5*C4	S60 = 0 + R7*C4
S13 = R2 + R1*C5	S29 = R4 + R3*C5	S45 = R6 + R5*C5	S61 = 0 + R7*C5
S14 = R2 + R1*C6	S30 = R4 + R3*C6	S46 = R6 + R5*C6	S62 = 0 + R7*C6
S15 = R2 + R1*C7	S31 = R4 + R3*C7	S47 = R6 + R5*C7	S63 = 0 + R7*C7
S16 = R2 + R1*0	S32 = R4 + R3*0	S48 = R6 + R5*0	S00 = 0 + R7*0

Now let us consider the mismatch scrambling circuit suitable for 2D layout as shown in Fig.4.10 and Table4-4. For each n (n=1, 2, 3,..,, 63, 64), we have

$$S_n = R'_{p+1} + R''_p C'_q$$

and p, q are changed dynamically in a pseudo random manner (p, q= 1, 2,..., 7, 8). Let us consider one-to-one mapping between $(C'_8,\ C'_7,\ C'_6,\ C'_5,\ C'_4,\ C'_3,\ C'_2,\ C'_1)$ and $(C_8,\ C_7,\ C_6,\ C_5,\ C_4,\ C_3,\ C_2,\ C_1)$, and their relationships are changed at each sampling time.

One example is as follows:

At time k:

$$(C'_8, C'_7, C'_6, C'_5, C'_4, C'_3, C'_2, C'_1) = (C_8, C_7, C_6, C_5, C_4, C_3, C_2, C_1)$$

At time k+1:

$$(C'_8, C'_7, C'_6, C'_5, C'_4, C'_3, C'_2, C'_1) = \quad (C_1, C_3, C_6, C_5, C_8, C_4, C_7, C_2)$$

At time k+2:

$$(C'_8, C'_7, C'_6, C'_5, C'_4, C'_3, C'_2, C'_1) = \quad (C_4, C_1, C_6, C_2, C_8, C_3, C_5, C_7)$$

Similarly, let us consider one-to-one mapping between $(R'_8,\ R'_7,\ R'_6,\ R'_5,\ R'_4,\ R'_3,\ R'_2,\ R'_1)$ and $(R_8,\ R_7,\ R_6,\ R_5,\ R_4,\ R_3,\ R_2,\ R_1)$, and we change their relationships dynamically. Also, the following relationship is maintained:

If $R'_{p+1} = R_{r+1}$, then $R"_p = R_r$ (p, r = 0, 1, 2, 3, 4, 5, 6, 7).

One example is as follows:

At time k:

$$(R'_8, R'_7, R'_6, R'_5, R'_4, R'_3, R'_2, R'_1) = \quad (R_8, R_7, R_6, R_5, R_4, R_3, R_2, R_1)$$

$$(R"_8, R"_7, R"_6, R"_5, R"_4, R"_3, R"_2, R"_1) = \quad (R_7, R_6, R_5, R_4, R_3, R_2, R_1, R_0)$$

At time k+1:

$$(R'_8, R'_7, R'_6, R'_5, R'_4, R'_3, R'_2, R'_1) = \quad (R_6, R_4, R_1, R_2, R_3, R_8, R_5, R_7)$$

$$(R"_8, R"_7, R"_6, R"_5, R"_4, R"_3, R"_2, R"_1) = \quad (R_5, R_3, R_0, R_1, R_2, R_7, R_4, R_6)$$

At time k+2:

$$(R'_8, R'_7, R'_6, R'_5, R'_4, R'_3, R'_2, R'_1) = \quad (R_5, R_1, R_6, R_2, R_8, R_3, R_7, R_4)$$

$$(R"_8, R"_7, R"_6, R"_5, R"_4, R"_3, R"_2, R"_1) = \quad (R_4, R_0, R_5, R_1, R_7, R_2, R_6, R_3)$$

These interconnection changes can be realized with multiplexer arrays (Row Scrambler, Column Scrambler in Fig.4.10) following the column and row decoders, as well as pseudo random signal generators (RNG_R, RNG_C in Fig.4.10) for control of their selection signals; RNG_R, RNG_C can be realized with linear feedback shift registers (LFSRs).

To understand the above mismatch scrambling for the 2D square layout, we explain the 4-bit unary DAC case for simplicity. Fig.4.11 (a) shows its 2D matrix layout, whereas Fig.4.11 (b) shows the one with scrambler circuits following the row and column decoders, and also it explains one example of the unit cell selection order change.

4.4 Mismatch Scrambling Technique for Virtual 3D Layout

The 2D layout mismatch scrambling algorithm in Section 3 may be enough for DACs with up-to 6-bit resolution. However, the authors often use 1D layout instead of 2D layout for a unary DAC with high resolution, e. g., 10-bit. In such a case, routing and scramblers using the 2D regular layout mismatch scrambling method becomes complicated and needs substantial chip area.

Then we propose here a virtual 3D layout mismatch scrambling circuit (Fig.4.12); this can reduce the total scrambling circuit amount and the number of inter-connections compared to the 2D layout mismatch scrambling circuit and hence this is more effective for higher resolution

DACs; the virtual 3D layout DAC can be done as 1D layout which currently we use for our DAC design.

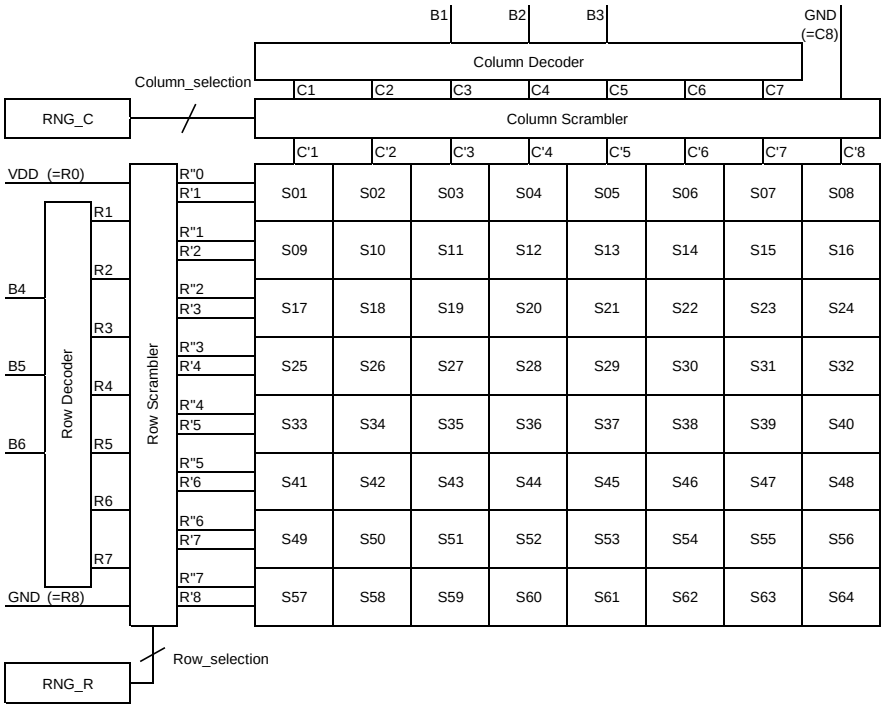


Fig. 4.10 2D unit cell layout of 6-bit unary DAC with mismatch scrambling.

Table 4-4 Logical expressions of unit cell local decoders for a 6-bit unary DAC with 2D layout mismatch scrambling in Fig.4.4.

S01 =	R'1 +	R"0* C'1	S17 =	R'3 +	R"2* C'1	S33 =	R'5 +	R"4* C'1	S49 =	R'7 +	R"6* C'1
S02 =	R'1 +	R"0* C'2	S18 =	R'3 +	R"2* C'2	S34 =	R'5 +	R"4* C'2	S50 =	R'7 +	R"6* C'2
S03 =	R'1 +	R"0* C'3	S19 =	R'3 +	R"2* C'3	S35 =	R'5 +	R"4* C'3	S51 =	R'7 +	R"6* C'3
S04 =	R'1 +	R"0* C'4	S20 =	R'3 +	R"2* C'4	S36 =	R'5 +	R"4* C'4	S52 =	R'7 +	R"6* C'4
S05 =	R'1 +	R"0* C'5	S21 =	R'3 +	R"2* C'5	S37 =	R'5 +	R"4* C'5	S53 =	R'7 +	R"6* C'5
S06 =	R'1 +	R"0* C'6	S22 =	R'3 +	R"2* C'6	S38 =	R'5 +	R"4* C'6	S54 =	R'7 +	R"6* C'6
S07 =	R'1 +	R"0* C'7	S23 =	R'3 +	R"2* C'7	S39 =	R'5 +	R"4* C'7	S55 =	R'7 +	R"6* C'7
S08 =	R'1 +	R"0* C'8	S24 =	R'3 +	R"2* C'8	S40 =	R'5 +	R"4* C'8	S56 =	R'7 +	R"6* C'8
S09 =	R'2 +	R"1* C'1	S25 =	R'4 +	R"3* C'1	S41 =	R'6 +	R"5* C'1	S57 =	R'8 +	R"7* C'1
S10 =	R'2 +	R"1* C'2	S26 =	R'4 +	R"3* C'2	S42 =	R'6 +	R"5* C'2	S58 =	R'8 +	R"7* C'2
S11 =	R'2 +	R"1* C'3	S27 =	R'4 +	R"3* C'3	S43 =	R'6 +	R"5* C'3	S59 =	R'8 +	R"7* C'3
S12 =	R'2 +	R"1* C'4	S28 =	R'4 +	R"3* C'4	S44 =	R'6 +	R"5* C'4	S60 =	R'8 +	R"7* C'4
S13 =	R'2 +	R"1* C'5	S29 =	R'4 +	R"3* C'5	S45 =	R'6 +	R"5* C'5	S61 =	R'8 +	R"7* C'5
S14 =	R'2 +	R"1* C'6	S30 =	R'4 +	R"3* C'6	S46 =	R'6 +	R"5* C'6	S62 =	R'8 +	R"7* C'6
S15 =	R'2 +	R"1* C'7	S31 =	R'4 +	R"3* C'7	S47 =	R'6 +	R"5* C'7	S63 =	R'8 +	R"7* C'7
S16 =	R'2 +	R"1* C'8	S32 =	R'4 +	R"3* C'8	S48 =	R'6 +	R"5* C'8	S64 =	R'8 +	R"7* C'8

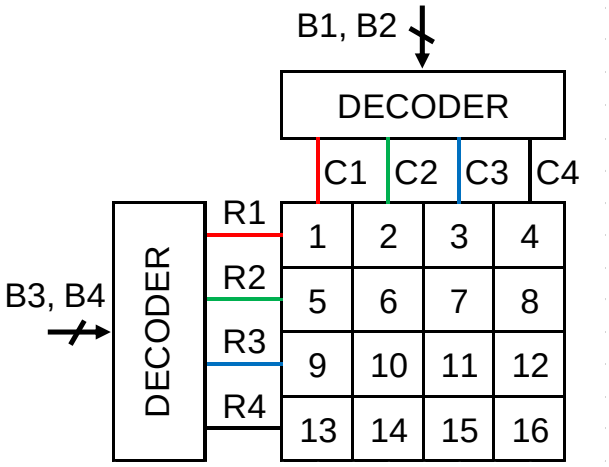


Fig. 4.11 (a) 4-bit unary DAC with 2D regular layout of unit cells.

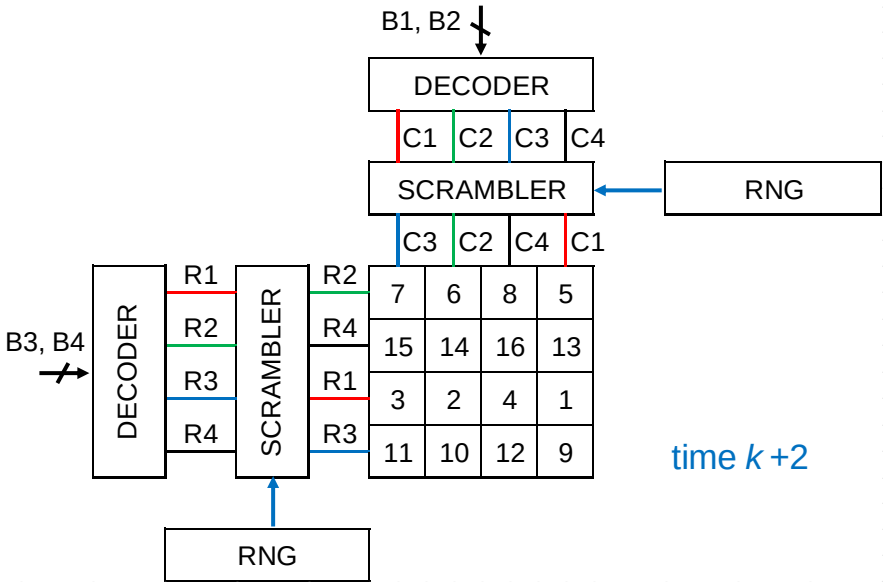


Fig. 4.11 (b) 4-bit unary DAC mismatch scrambling circuit and operation example.

Let us consider a 6-bit unary DAC with digital binary inputs of B_6, B_5, B_4, B_3, B_2, B_1, and their thermometer codes are S_{64}, S_{63}, S_{62}, …, S_3, S_2,

S_1. Notice that its extension to a higher resolution unary DAC is straightforward.

First, the concept of the virtual 3D layout of a unary DAC using X, Y, Z decoders without mismatch scrambling is shown in Fig.4.12 and its actual 1D layout implementation is in Fig.4.13 There, 3 binary-to-thermometer code decoders with truth tables in Table4-5 are used.

Z-decoder: binary inputs $B_6,\ B_5$

Thermometer outputs $Z_3,\ Z_2,\ Z_1$.

$$Z_3{=}B_6B_5,\ Z_2{=}B_6,\ Z_1{=}B_6{+}B_5.$$

ii) Y-decoder: binary inputs $B_4,\ B_3$

Thermometer outputs $Y_3,\ Y_2,\ Y_1$.

$$Y_3{=}B_4B_3,\ Y_2{=}B_4,\ Y_1{=}B_4{+}B_3.$$

iii) X-decoder: binary inputs $B_2,\ B_1$

Thermometer outputs $X_3,\ X_2,\ X_1$.

$$X_3{=}B_2B_1,\ X_2{=}B_2,\ X_1{=}B_2{+}B_1.$$

The followings are set:

$$Z_0{=}Y_0{=}X_0{=}1,\ Z_4{=}Y_4{=}X_4{=}0.$$

The local decoder for n-th cell is given as follows (Table 4-6).

Here values of a, b, c are given as follows:

$$\text{For}\quad 1 \le n \le 16, \qquad a = 0.$$

For $17 \leq n \leq 32$, $a = 1$.

For $33 \leq n \leq 48$, $a = 2$.

For $49 \leq n \leq 64$, $a = 3$

For $1 \leq n \leq 4$, $17 \leq n \leq 20$, $33 \leq n \leq 36$,

or $49 \leq n \leq 52$, $b = 0$.

For $5 \leq n \leq 8$, $21 \leq n \leq 24$, $37 \leq n \leq 40$,

or $53 \leq n \leq 56$, $b = 1$.

For $9 \leq n \leq 12$, $25 \leq n \leq 28$, $41 \leq n \leq 44$,

or $57 \leq n \leq 60$, $b = 2$.

For $13 \leq n \leq 16$, $29 \leq n \leq 32$, $45 \leq n \leq 48$,

or $61 \leq n \leq 64$, $b = 3$.

For $1 \leq \mathrm{mod}_4\ (n) \leq 3$, $c = \mathrm{mod}_4\ (n)$.

For $\mathrm{mod}_c(n) = 0$, $c = 4$.

Next, we show our proposed virtual 3D layout mismatch scrambling in Fig.4.14. There, scramblers or multiplexer arrays with selection signal control circuits of LFSRs (RNG_X, RNG_Y, RNG_Z) are added.

Scrambler_X:

inputs $(X_4,\ X_3,\ X_2,\ X_1)$

outputs $(X'_4,\ X'_3,\ X'_2,\ X'_1)$

Scrambler_Y:

inputs $(Y_4,\ Y_3,\ Y_2,\ Y_1,\ Y_0)$

outputs $(Y'_4,\ Y'_3,\ Y'_2,\ Y'_1,\ Y"_3,\ Y"_2,\ Y"_1,\ Y"_0)$

Also, the following relationship is maintained:

If $Y'_{p+1}= Y_{n+1}$, then $Y"_p= Y_n$ (p, n = 0, 1, 2, 3).

Scrambler_Z:

inputs $(Z_4,\ Z_3,\ Z_2,\ Z_1,\ Z_0)$

outputs $(Z'_4,\ Z'_3,\ Z'_2,\ Z'_1,\ Z"_3,\ Z"_2,\ Z"_1,\ Z"_0)$

Also, the following relationship is maintained:

If $Z'_{p+1}= Z_{n+1}$, then $Z"_p= Z_n$ (p, n = 0, 1, 2, 3).

These inputs and outputs are one-to-one mappings, and have the following inter-connections (Table 4-6):

S_n=Z'_{n+1}+$Z"_n Y'_{m+1} + Z"_n Y"_m X'_l$

Let us consider one-to-one mapping between (X'_4, X'_3, X'_2, X'_1) and (X_4, X_3, X_2, X_1), and their relationships are changed dynamically (Fig.4.14, Fig4.15). This structure can reduce the circuit size and wiring complexity of the decoder.

One example is as follows:

At time k:

$$(X'_4, X'_3, X'_2, X'_1) = \ (X_4, X_3, X_2, X_1)$$

At time k+1:

$$(X'_4, X'_3, X'_2, X'_1) = (X_1, X_3, X_2, X_4)$$

At time k+2:

$$(X'_4, X'_3, X'_2, X'_1) = (X_3, X_4, X_1, X_2)$$

At time k+3:

$$(X'_4, X'_3, X'_2, X'_1) = (X_2, X_1, X_4, X_3)$$

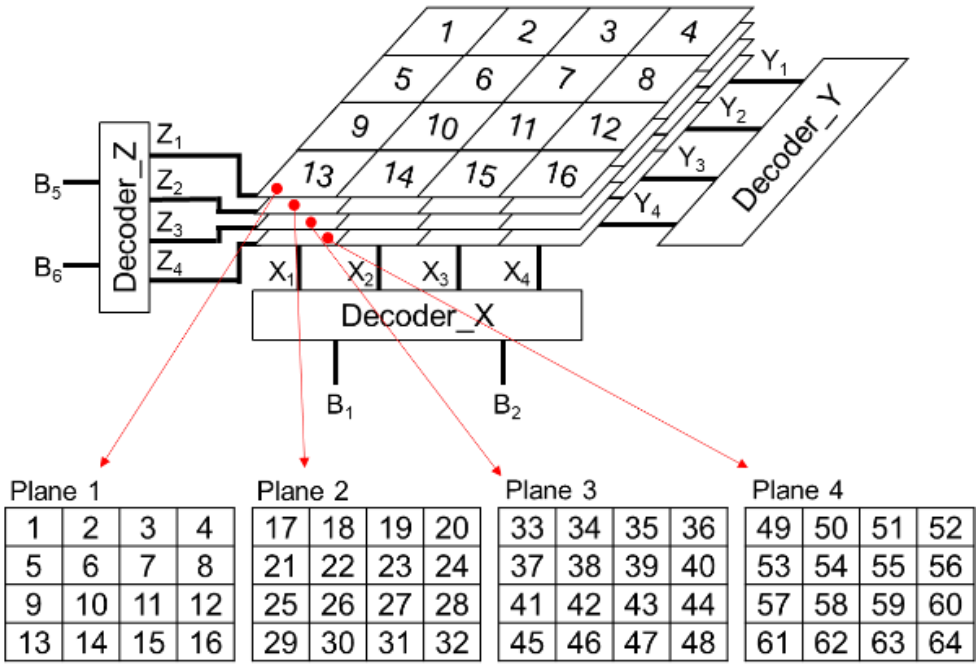


Fig. 4.12 Virtual 3D layout of 6-bit unary DAC.

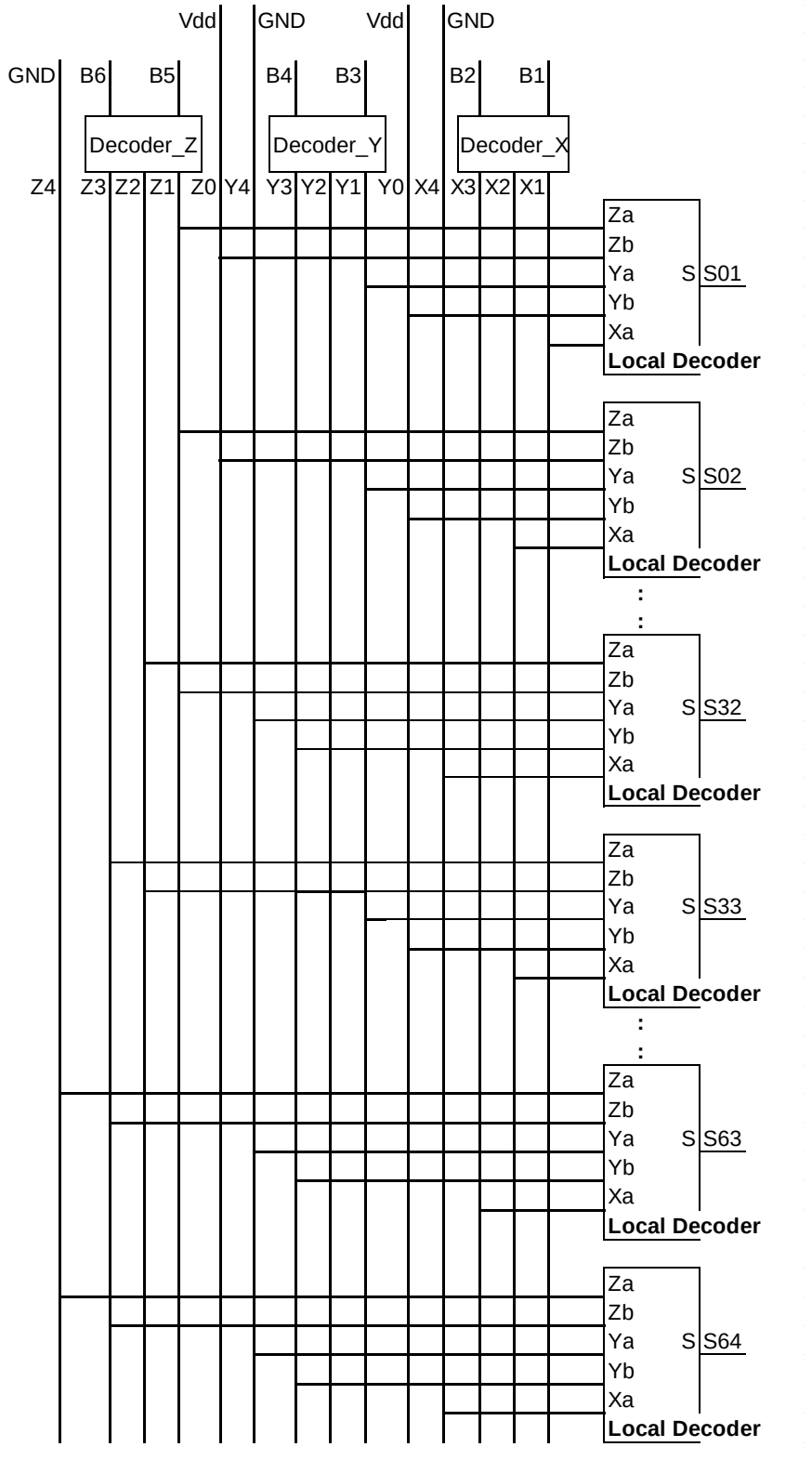


Fig. 4.13 1D unit cell layout of 6-bit unary DAC with virtual 3D layout.

Table 4-5 Truth tables of binary-to-thermometer decoders for upper 2-bit (left), middle 2-bit (middle) and lower 2-bit (right) of a 6-bit unary DAC.

B6	B5	Z3	Z2	Z1
0	0	0	0	0
0	1	0	0	1
1	0	0	1	1
1	1	1	1	1

B4	B3	Y3	Y2	Y1
0	0	0	0	0
0	1	0	0	1
1	0	0	1	1
1	1	1	1	1

B2	B1	X3	X2	X1
0	0	0	0	0
0	1	0	0	1
1	0	0	1	1
1	1	1	1	1

Similarly, let us consider one-to-one mapping between $(Y'_4,\ Y'_3,\ Y'_2,\ Y'_1)$ and $(Y_4,\ Y_3,\ Y_2,\ Y_1)$, and we change their relationships dynamically.

One example is as follows:

At time *k*:

$$(Y'_4,\ Y'_3,\ Y'_2,\ Y'_1) = (Y_4, Y_3, Y_2, Y_1)$$

$$(Y"_4,\ Y"_3,\ Y"_2,\ Y"_1) = (Y_3, Y_2, Y_1, Y_0)$$

At time *k*+1:

$$(Y'_4,\ Y'_3,\ Y'_2,\ Y'_1) = (Y_3, Y_2, Y_1, Y_4)$$

$$(Y"_4,\ Y"_3,\ Y"_2,\ Y"_1) = (Y_2, Y_1, Y_0, Y_3)$$

At time *k*+2:

$$(Y'_4,\ Y'_3,\ Y'_2,\ Y'_1) = (Y_2, Y_1, Y_4, Y_3)$$

$$(Y"_4,\ Y"_3,\ Y"_2,\ Y"_1) = (Y_1, Y_0, Y_3, Y_2)$$

At time *k*+3:

$$(Y'_4,\ Y'_3,\ Y'_2,\ Y'_1) = (Y_1, Y_2, Y_3, Y_4)$$

$$(Y"_4,\ Y"_3,\ Y"_2,\ Y"_1) \quad = \quad (Y_0, Y_1, Y_2, Y_3)$$

Also, consider one-to-one mapping between$(Z'_4,\ Z'_3,\ Z'_2,\ Z'_1)$ and $(Z_4,\ Z_3,\ Z_2,\ Z_1)$, and their relationships are changed dynamically. One example is as follows:

At time *k*:

$$(Z'_4,\ Z'_3,\ Z'_2,\ Z'_1) = \quad (Z_4, Z_3, Z_2, Z_1)$$

$$(Z"_4,\ Z"_3,\ Z"_2,\ Z"_1) = \quad (Z_3, Z_2, Z_1, Z_0)$$

At time *k*+1:

$$(Z'_4,\ Z'_3,\ Z'_2,\ Z'_1) = \quad (Z_2, Z_3, Z_1, Z_4)$$

$$(Z"_4,\ Z"_3,\ Z"_2,\ Z"_1) = \quad (Z_1, Z_2, Z_0, Z_3)$$

At time *k*+2:

$$(Z'_4,\ Z'_3,\ Z'_2,\ Z'_1) = \quad (Z_3, Z_4, Z_1, Z_2)$$

$$(Z"_4,\ Z"_3,\ Z"_2,\ Z"_1) = \quad (Z_2, Z_3, Z_0, Z_1)$$

At time *k*+3:

$$(Z'_4,\ Z'_3,\ Z'_2,\ Z'_1) = \quad (Z_2, Z_1, Z_4, Z_3)$$

$$(Z"_4,\ Z"_3,\ Z"_2,\ Z"_1) = \quad (Z_1, Z_0, Z_3, Z_2)$$

Table 4-6 Logical expressions of unit cell local decoders for a 6-bit unary DAC with virtual 3D layout in Fig.4. 5.

S01 = Z1 + Z0*Y1 + Z0*Y0*X1	S33 = Z3 + Z2*Y1 + Z2*Y0*X1
S02 = Z1 + Z0*Y1 + Z0*Y0*X2	S34 = Z3 + Z2*Y1 + Z2*Y0*X2
S03 = Z1 + Z0*Y1 + Z0*Y0*X3	S35 = Z3 + Z2*Y1 + Z2*Y0*X3
S04 = Z1 + Z0*Y1 + Z0*Y0*X4	S36 = Z3 + Z2*Y1 + Z2*Y0*X4
S05 = Z1 + Z0*Y2 + Z0*Y1*X1	S37 = Z3 + Z2*Y2 + Z2*Y1*X1
S06 = Z1 + Z0*Y2 + Z0*Y1*X2	S38 = Z3 + Z2*Y2 + Z2*Y1*X2
S07 = Z1 + Z0*Y2 + Z0*Y1*X3	S39 = Z3 + Z2*Y2 + Z2*Y1*X3
S08 = Z1 + Z0*Y2 + Z0*Y1*X4	S40 = Z3 + Z2*Y2 + Z2*Y1*X4
S09 = Z1 + Z0*Y3 + Z0*Y2*X1	S41 = Z3 + Z2*Y3 + Z2*Y2*X1
S10 = Z1 + Z0*Y3 + Z0*Y2*X2	S42 = Z3 + Z2*Y3 + Z2*Y2*X2
S11 = Z1 + Z0*Y3 + Z0*Y2*X3	S43 = Z3 + Z2*Y3 + Z2*Y2*X3
S12 = Z1 + Z0*Y3 + Z0*Y2*X4	S44 = Z3 + Z2*Y3 + Z2*Y2*X4
S13 = Z1 + Z0*Y4 + Z0*Y3*X1	S45 = Z3 + Z2*Y4 + Z2*Y3*X1
S14 = Z1 + Z0*Y4 + Z0*Y3*X2	S46 = Z3 + Z2*Y4 + Z2*Y3*X2
S15 = Z1 + Z0*Y4 + Z0*Y3*X3	S47 = Z3 + Z2*Y4 + Z2*Y3*X3
S16 = Z1 + Z0*Y4 + Z0*Y3*X4	S48 = Z3 + Z2*Y4 + Z2*Y3*X4
S17 = Z2 + Z1*Y1 + Z1*Y0*X1	S49 = Z4 + Z3*Y1 + Z3*Y0*X1
S18 = Z2 + Z1*Y1 + Z1*Y0*X2	S50 = Z4 + Z3*Y1 + Z3*Y0*X2
S19 = Z2 + Z1*Y1 + Z1*Y0*X3	S51 = Z4 + Z3*Y1 + Z3*Y0*X3
S20 = Z2 + Z1*Y1 + Z1*Y0*X4	S52 = Z4 + Z3*Y1 + Z3*Y0*X4
S21 = Z2 + Z1*Y2 + Z1*Y1*X1	S53 = Z4 + Z3*Y2 + Z3*Y1*X1
S22 = Z2 + Z1*Y2 + Z1*Y1*X2	S54 = Z4 + Z3*Y2 + Z3*Y1*X2
S23 = Z2 + Z1*Y2 + Z1*Y1*X3	S55 = Z4 + Z3*Y2 + Z3*Y1*X3
S24 = Z2 + Z1*Y2 + Z1*Y1*X4	S56 = Z4 + Z3*Y2 + Z3*Y1*X4
S25 = Z2 + Z1*Y3 + Z1*Y2*X1	S57 = Z4 + Z3*Y3 + Z3*Y2*X1
S26 = Z2 + Z1*Y3 + Z1*Y2*X2	S58 = Z4 + Z3*Y3 + Z3*Y2*X2
S27 = Z2 + Z1*Y3 + Z1*Y2*X3	S59 = Z4 + Z3*Y3 + Z3*Y2*X3
S28 = Z2 + Z1*Y3 + Z1*Y2*X4	S60 = Z4 + Z3*Y3 + Z3*Y2*X4
S29 = Z2 + Z1*Y4 + Z1*Y3*X1	S61 = Z4 + Z3*Y4 + Z3*Y3*X1
S30 = Z2 + Z1*Y4 + Z1*Y3*X2	S62 = Z4 + Z3*Y4 + Z3*Y3*X2
S31 = Z2 + Z1*Y4 + Z1*Y3*X3	S63 = Z4 + Z3*Y4 + Z3*Y3*X3
S32 = Z2 + Z1*Y4 + Z1*Y3*X4	S64 = Z4 + Z3*Y4 + Z3*Y3*X4

X0=Y0=Z0=1 X4=Y4=Z4=0

4.5 Simulation Verification

We have simulated a 6-bit unary DAC with some unit current mismatches and compared the results without mismatch scrambling, with the one for 2D regular layout and with the proposed one for virtual 3D layout. Their output power spectrum and SFDRs are shown in Fig.4.16. We see that

both mismatch scrambling methods improve the SFDR compared to the case without mismatch scrambling, and both mismatch scrambling effects are comparable.

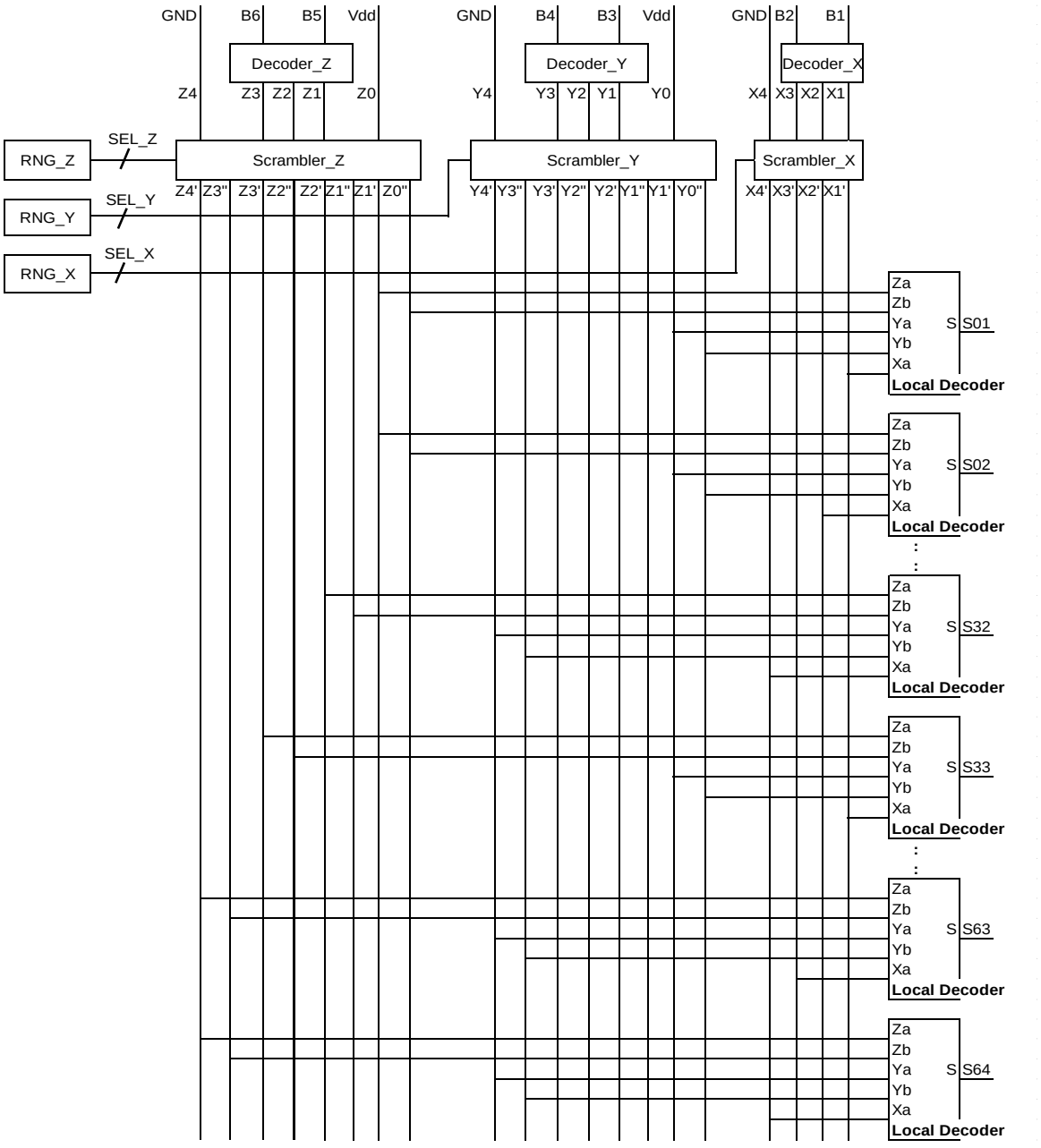


Fig. 4.14 1D unit cell layout of 6-bit unary DAC for virtual 3D layout mismatch scrambling.

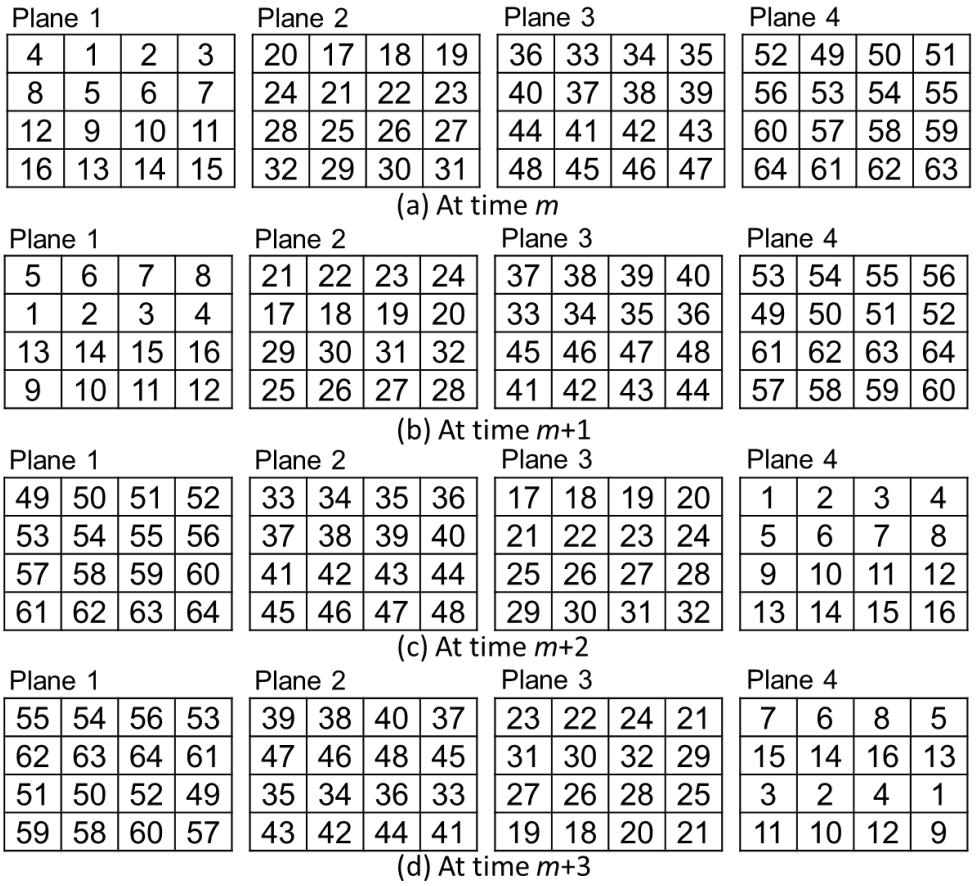


Fig. 4.15 One example of mismatch scrambling operation of a 6-bit unary DAC with virtual 3D layout in Fig.4. 5

Table 4-7 Logical expressions of unit cell local decoders for a 6-bit unary DAC with virtual 3D layout mismatch scrambling in Fig. 4.14.

S01 =	Z1'+	Z0"*Y1'+	Z0"*Y0"*X1'	S33 =	Z3'+	Z2"*Y1'+	Z2"*Y0"*X1'
S02 =	Z1'+	Z0"*Y1'+	Z0"*Y0"*X2'	S34 =	Z3'+	Z2"*Y1'+	Z2"*Y0"*X2'
S03 =	Z1'+	Z0"*Y1'+	Z0"*Y0"*X3'	S35 =	Z3'+	Z2"*Y1'+	Z2"*Y0"*X3'
S04 =	Z1'+	Z0"*Y1'+	Z0"*Y0"*X4'	S36 =	Z3'+	Z2"*Y1'+	Z2"*Y0"*X4'
S05 =	Z1'+	Z0"*Y2'+	Z0"*Y1"*X1'	S37 =	Z3'+	Z2"*Y2'+	Z2"*Y1"*X1'
S06 =	Z1'+	Z0"*Y2'+	Z0"*Y1"*X2'	S38 =	Z3'+	Z2"*Y2'+	Z2"*Y1"*X2'
S07 =	Z1'+	Z0"*Y2'+	Z0"*Y1"*X3'	S39 =	Z3'+	Z2"*Y2'+	Z2"*Y1"*X3'
S08 =	Z1'+	Z0"*Y2'+	Z0"*Y1"*X4'	S40 =	Z3'+	Z2"*Y2'+	Z2"*Y1"*X4'
S09 =	Z1'+	Z0"*Y3'+	Z0"*Y2"*X1'	S41 =	Z3'+	Z2"*Y3'+	Z2"*Y2"*X1'
S10 =	Z1'+	Z0"*Y3'+	Z0"*Y2"*X2'	S42 =	Z3'+	Z2"*Y3'+	Z2"*Y2"*X2'
S11 =	Z1'+	Z0"*Y3'+	Z0"*Y2"*X3'	S43 =	Z3'+	Z2"*Y3'+	Z2"*Y2"*X3'
S12 =	Z1'+	Z0"*Y3'+	Z0"*Y2"*X4'	S44 =	Z3'+	Z2"*Y3'+	Z2"*Y2"*X4'
S13 =	Z1'+	Z0"*Y4'+	Z0"*Y3"*X1'	S45 =	Z3'+	Z2"*Y4'+	Z2"*Y3"*X1'
S14 =	Z1'+	Z0"*Y4'+	Z0"*Y3"*X2'	S46 =	Z3'+	Z2"*Y4'+	Z2"*Y3"*X2'
S15 =	Z1'+	Z0"*Y4'+	Z0"*Y3"*X3'	S47 =	Z3'+	Z2"*Y4'+	Z2"*Y3"*X3'
S16 =	Z1'+	Z0"*Y4'+	Z0"*Y3"*X4'	S48 =	Z3'+	Z2"*Y4'+	Z2"*Y3"*X4'
S17 =	Z2'+	Z1"*Y1'+	Z1"*Y0"*X1'	S49 =	Z4'+	Z3"*Y1'+	Z3"*Y0"*X1'
S18 =	Z2'+	Z1"*Y1'+	Z1"*Y0"*X2'	S50 =	Z4'+	Z3"*Y1'+	Z3"*Y0"*X2'
S19 =	Z2'+	Z1"*Y1'+	Z1"*Y0"*X3'	S51 =	Z4'+	Z3"*Y1'+	Z3"*Y0"*X3'
S20 =	Z2'+	Z1"*Y1'+	Z1"*Y0"*X4'	S52 =	Z4'+	Z3"*Y1'+	Z3"*Y0"*X4'
S21 =	Z2'+	Z1"*Y2'+	Z1"*Y1"*X1'	S53 =	Z4'+	Z3"*Y2'+	Z3"*Y1"*X1'
S22 =	Z2'+	Z1"*Y2'+	Z1"*Y1"*X2'	S54 =	Z4'+	Z3"*Y2'+	Z3"*Y1"*X2'
S23 =	Z2'+	Z1"*Y2'+	Z1"*Y1"*X3'	S55 =	Z4'+	Z3"*Y2'+	Z3"*Y1"*X3'
S24 =	Z2'+	Z1"*Y2'+	Z1"*Y1"*X4'	S56 =	Z4'+	Z3"*Y2'+	Z3"*Y1"*X4'
S25 =	Z2'+	Z1"*Y3'+	Z1"*Y2"*X1'	S57 =	Z4'+	Z3"*Y3'+	Z3"*Y2"*X1'
S26 =	Z2'+	Z1"*Y3'+	Z1"*Y2"*X2'	S58 =	Z4'+	Z3"*Y3'+	Z3"*Y2"*X2'
S27 =	Z2'+	Z1"*Y3'+	Z1"*Y2"*X3'	S59 =	Z4'+	Z3"*Y3'+	Z3"*Y2"*X3'
S28 =	Z2'+	Z1"*Y3'+	Z1"*Y2"*X4'	S60 =	Z4'+	Z3"*Y3'+	Z3"*Y2"*X4'
S29 =	Z2'+	Z1"*Y4'+	Z1"*Y3"*X1'	S61 =	Z4'+	Z3"*Y4'+	Z3"*Y3"*X1'
S30 =	Z2'+	Z1"*Y4'+	Z1"*Y3"*X2'	S62 =	Z4'+	Z3"*Y4'+	Z3"*Y3"*X2'
S31 =	Z2'+	Z1"*Y4'+	Z1"*Y3"*X3'	S63 =	Z4'+	Z3"*Y4'+	Z3"*Y3"*X3'
S32 =	Z2'+	Z1"*Y4'+	Z1"*Y3"*X4'	S64 =	Z4'+	Z3"*Y4'+	Z3"*Y3"*X4'

X0"=Y0"=Z0"=1 X4'=Y4'=Z4'=0

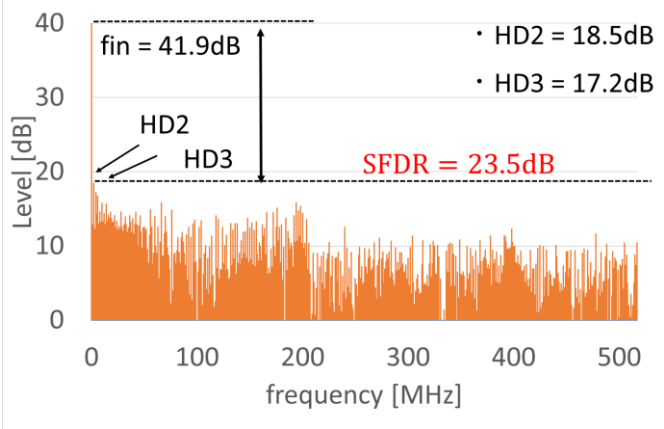


Fig. 4.16 (a) Simulated 6-bit DAC output power spectrum.

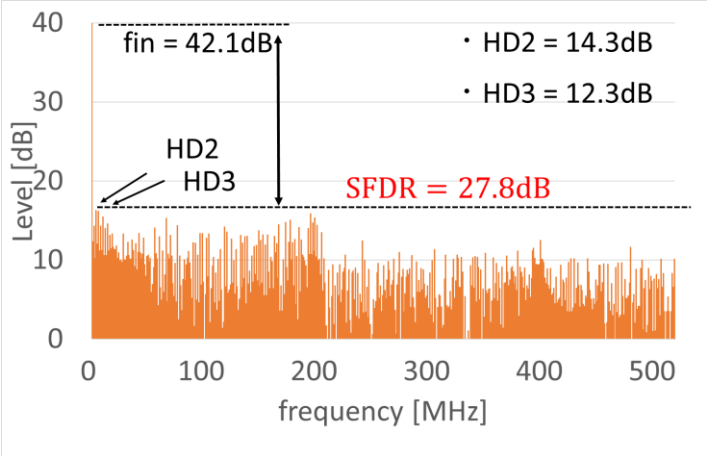


Fig. 4.16 (b) Without mismatch scrambling.

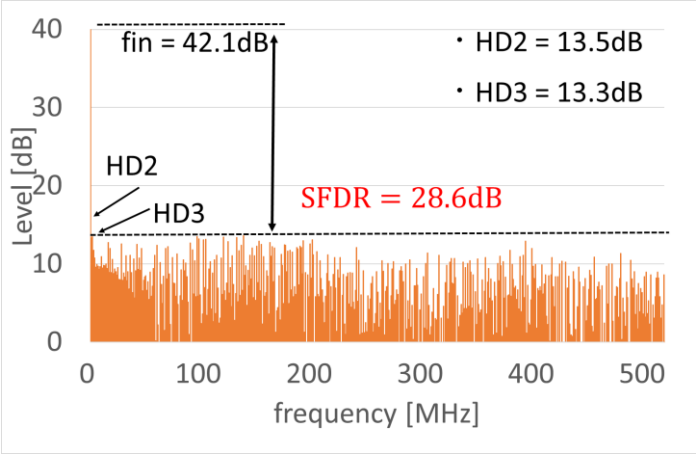


Fig. 4.16 (c) With mismatch scrambling for virtual 3D layout.

4.6 Summary

4.6.1Conclusion

In this dissertation, we have proposed unary DAC mismatch scrambling circuit based on virtual 3D layout concept, but there 1D layout is actually used. Its circuit implementation requires only small amount of circuits compared to the conventional methods. Simulation results show that its SFDR can be improved compared to the case without the mismatch scrambling.

4.6.2 Items for the Future Study

From the above simulation results, we can see that 3D has improved SFDR, so can we expand to virtual 4D or higher dimensional layout.

It is also possible to carry out circuit simulation to verify whether these methods can really be improved. If the verification is successful, it can be tested in the circuit by making a real object.

References

[30] F. Maloberti: Data converters, Spring (2007).

[31] R. V. D. Plassche: Integrated analog-to-digital and digital-to-analog converters, Springer (2012).

[32] I. Galton: “Why dynamic-element-matching DACs Work”, IEEE TCAS-II 57 (2010).

[33] M. Kilic, et al: “A row-column accessed dynamic element matching DAC architecture for SAR ADCs”, IEEE Nordic Circuits and Systems Conference and International Symposium of System-on-Chip (2018).

[34] T. Miki, et al.: "An 80-MHz 8-bit CMOS DAC”, IEEE Journal of Solid-State Circuits (Dec. 1986).

[35] M. J. M. Pelgrom, et al.: “Matching properties of MOS transistors”, IEEE Journal of Solid-State Circuits (1989).

[36] N. U. Andersson, et al.: “Models and implementation of a dynamic element matching DAC”, Analog Integrated Circuits and Signal Processing, Springer (2003).

[37] H. Kobayashi, et al.: “Performance improvement of delta-sigma ADC/DAC/TDC using digital technique”, IEEE ICSICT (2018).

[38] R. T. Baird et al. : “Linearity enhancement of multibit ΔΣ A/D and DACs using data weighted averaging”, IEEE TCAS-II (1995).

[39] J. Remple, et al.: “An ISI scrambling technique for dynamic element matching current steering DACs”, IEEE Journal of Solid-State Circuits (2022).

[40] J. Liu, et al.: “A 14-bit 1.0-GS/s dynamic element matching DAC with >80 dB SFDR up to the Nyquist”, IEEE ISCAS (2015).

[41] B. Liu, et al.: “A novel dynamic element match technique in current-steering DAC”, IEEE ASICON (2013).

[42] K. Nguyen: “Practical dynamic element matching techniques for 3-level unit elements”, IEEE CICC (2017).

[43] D.-H. Lee, et al.: “Low-cost 14-bit current-steering DAC with a randomized thermometer-coding method”, IEEE TCAS-II 56 (2009).

[44] N. U. Andersson and J. J. Wikner: “A strategy for implementing dynamic element matching in current-steering DACs”, Southwest Symposium on Mixed-Signal Design (2002).

[45] S. Huang, et al., “A 14b 1GS/s DAC with SFDR > 80 dBc across the whole Nyquist band by mixed total 3-dimesional sort-and-combine and dynamic element matching”, IEEE ASICON (2015)

[46] J. Kojima, et al.: “DWA algorithm for band-pass ΔΣ DAC with ternary unit cells”, IEEE ICSICT (2018).

[47] D.-H. Yoon, et al., “LW-DEM: designing a low power digital-to-analog converter using lightweight dynamic element matching technique”, IEEE Access (2019).

[48] D.-H. Kee, et al.: “Nyquist-rate current-steering digital-to-analog converters with random multiple data-weighted averaging technique and QN rotated walk switching scheme”, IEEE TCAS-II (2006).

[49] W. -T. Lin, T.-H. Kuo: “A compact dynamic-performance-improved current-steering DAC with random rotation-based binary-weighted selection”, IEEE Journal of Solid-State Circuits 47 (2012).

[50] A. K. Gupta, et al.: “Second order dynamic element matching technique for low oversampling delta sigma ADC”, IEEE ISCAS (2006).

[51] Y. Tang, et al.: “A 14b 200MS/s DAC with SFDR>78dBc, IM3<−83dBc and NSD<−163dBm/Hz across the whole Nyquist band enabled by dynamic-mismatch mapping”, VLSI Circuits Symp. (2010).

[52] L. Lai, et al.: “Demystifying and mitigating code-dependent switching distortions in current-steering DACs”, IEEE TCAS-I (2019).

[53] H. Jiang, et al.: “A segmented thermometer coded DAC with deterministic dynamic element matching for high resolution ADC test”, IEEE ISCAS (2005).

[54] J. W. Bruce, P. Stubberud: “Generalized cube networks for implementing dynamic element matching digital-to-analog converters”, IEEE MWSCAS(1998).

[55] Z. Li, T. S. Fiez: “Dynamic element matching in low oversampling delta sigma ADCs”, IEEE ISCAS (2002).

[56] J. O. Coleman: “Mathematical unification of dynamic-element-matching methods for spectral shaping of hardware-mismatch errors”, IEEE MWSCAS (2000).

[57] M. Vesterbacka: “Dynamic element matching in DACs with restricted scrambling”, IEEE ICECS (Dec. 2000).

[58] M. K. Rudberg, et al.: “Glitch minimization and dynamic element matching in DACs”, IEEE ICECS (Dec. 2000).

[59] H. San, et al.:“A noise-shaping algorithm of multi-bit DAC nonlinearities in complex bandpass ΔΣAD modulators", IEICE Trans. Fundamentals (April 2004).

[60] D. Yao, X. Bai, A. Kuwana, K. Kawauchi, M. Higashino, H. Kobayashi, A. Suzuki, S. Yamada, T. Kato, N. Ono, K. Miura, K. Hirai, R. Kitakoga, “Segmented DAC Unit Cell Selection Algorithm and Layout/Routing Based on Euler’s Knight Tour”, International Conference on Analog VLSI Circuits, Bordeaux, France (Oct. 2021).

Chapter 5

Conclusions and Future Work

5.1 Conclusion

Digital-to-analog converters are important modules in signal processing systems, which are widely used in military and civilian equipment and have broad market prospects. The vigorous development of the mobile Internet has promoted the low power consumption of handheld electronic products. Therefore, it is of great significance to design a low-power DAC to promote the development of mobile terminals.

We have demonstrated that basing the layout algorithm on the classical mathematics can improve segmented DAC linearity as it well systematic mismatch effects. Pseudo random numbers in 2D arrays were simulated to reproduce by the cell arrangements yielded by the regular, Latin Square, Magic Square and Euler Knight Tour approaches. Our simulations of several cases showed that the Euler Knight Tour offers equal or superior overall DAC linearity to the Latin Square and Magic Square techniques. We 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 techniques with useful properties and interesting

mathematical research results, layout algorithms using them to configure 2D unit cells of the unary DAC can be refined further.

We have proposed unary DAC mismatch scrambling circuit based on virtual 3D layout concept, but there 1D layout is actually used. Its circuit implementation requires only small amount of circuits compared to the conventional methods. Simulation results show that its SFDR can be improved compared to the case without the mismatch scrambling.

In a word, the research on this topic has enabled me to master the circuit of DAC, especially the design method of unary DAC and segmented DAC, which has improved my scientific research ability and laid the foundation for further research and application work in the future.

5.2 Future Work

In the above simulation, excel, C language and MATLAB are used. The software used is very complicated. If it is possible, I want to unify one language for simulation.

After that, I want to use the method proposed in the dissertation in FPGA for simulation verification and review. If it can want to apply to the actual circuit.

In the DAC, I only studied and improved a small part of the circuit, and I will learn other parts later to see if the algorithm proposed above can be applied in other parts.

List of Published Papers

Journal Papers

[1] **Dan Yao**, Xuanyan Bai, Shogo Katayama, Anna Kuwana, Kazuyuki Kawauchi, Haruo Kobayashi,Kouji Hirai, Akira Suzuki, Satoshi Yamada, Tomoyuki Kato, Ritsuko Kitakoga, Takeshi Shimamura, Gopal Adhikari, Nobuto Ono, Kazuhiro Miura, Shigeya Yamaguchi"Unit Cell Mismatch Scrambling Method for High-Resolution Unary DAC based on Virtual 3D Layout" IEICE Electronics Express, Volume 19, Issue 24, Pages 20220430, (Advance online publication October 13, 2022), (Published: December 25, 2022).

[2] **Dan Yao**, Xueyan Bai, Anna Kuwana, Kazuyuki Kawauchi, Yujie Zhao, Jianglin Wei, Shogo Katayama, Masashi Higashino, Haruo Kobayashi"Segmented DAC Unit Cell Selection Algorithm and Layout/Routing Based on Classical Mathematics" Journal of Mechanical and Electrical Intelligent System (JMEIS, J. Mech. Elect. Intel. Syst.), Vol.6, No.1, pp.13-30, (31st January 2023)..

[3] Xueyan Bai, Shogo Katayama, Dan Yao, Anna Kuwana, Zifei Xu, Haruo Kobayashi,"Asynchronous Capacitive SAR ADC based on Hopfield Network", IEICE Electronics Express, Vol. 19 No. 18. Pages 20220276 (Advance online publication: August 09, 2022) (Published: September 25th, 2022)

[4] Xueyan Bai, **Dan Yao**, Yuanyang Du, Minh Tri Tran, Shogo Katayama, Jianglin Wei, Yujie Zhao, Anna Kuwana, Haruo Kobayashi, Kazuyoshi Kubo"Derivation of digital-to-analog converter architectures based on number theory"Journal of Pure and Applied Mathematics, Volume 6, Issue 5,

(Published: October 30, 2022).

[5] Masashi Higashino, Shaiful Nizam Bin Mohyar, **Yao Dan**, Yifei Sun, Anna Kuwana, Haruo 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 (Jan. 2020).

[6] Anna Kuwana, Xueyan Bai, **Dan Yao**, Haruo Kobayashi,"Numerical Simulation for the Starting Characteristics of a Wind Turbine", Advanced Engineering Forum Vol. 38, pp.215-221, (Nov. 2020).

International Conference Papers

[1] Haruo Kobayashi, Kentaroh Katoh, Shuhei Yamamoto, Yujie Zhao, Shogo Katayama Jianglin Wei, Yonglun Yan, **Dan Yao**, Xueyan Bai, Anna Kuwana,"Challenges for Waveform Sampling and Related Technologies"2022 IEEE 16th International Conference on Solid-State and Integrated Circuit Technology (ICSICT)Nanjing, China, virtually on line, (Oct. 25-28, 2022)

[2] **Dan Yao**, Anna Kuwana, Haruo Kobayashi, Kazuyuki Kawauchi,"Digital-to-Analog Converter Linearity Improvement Technique Based on Classical Number Theory for Modern ULSI"30th International Workshop on Post-Binary ULSI Systems,May 28th, 2021, Fully Virtual

[3] **Dan Yao**, Xueyan Bai, Anna Kuwana, Kazuyuki Kawauchi, Masashi Higashino, Haruo Kobayashi, Akira Suzuki, Satoshi Yamada, Tomoyuki Kato, Nobuto Ono, Kazuhiro Miura, Kouji Hirai, Ritsuko Kitakoga,"Segmented DAC Unit Cell Selection Algorithm and Layout/Routing Based on Euler's Knight Tour" International Conference on Analog VLSI Circuits (AVIC 2021),Bordeaux, France (Oct. 18-21, 2021)

[4] Xueyan Bai, **Dan Yao**, Yuanyang Du, Minh Tri Tran, Anna Kuwana, Haruo Kobayashi, Kazuyoshi Kubo,"Design of Digital-to-Analog Converter Architectures Based on Polygonal Numbers" International Conference on Analog VLSI Circuits (AVIC 2021),Bordeaux, France (Oct. 18-21, 2021)

[5] Haruo Kobayashi, Xueyan Bai, Yujie Zhao, Shuhei Yamamoto, **Dan Yao**, Manato Hirai, Jianglin Wei, Shogo Katayama, Anna Kuwana,"Classical Mathematics and Analog/Mixed-Signal IC Design" IEEE 14th International Conference on ASIC (ASICON 2021)

[6] Zifei Xu, Xueyan Bai, **Dan Yao**, Anna Kuwana, Haruo

Kobayashi,"Revisit to Hopfield Network for Asynchronous SAR ADC and DAC", IEEE 3rd International Conference on Circuits and Systems (IEEE ICCS 2021),Chengdu, China (Oct. 29-31, 2021)

[7] Haruo Kobayashi, Yuto Sasaki, Hirotaka Arai, **Dan Yao**, Yujie Zhao, Xueyan Bai,Anna Kuwana,"Unified Methodology of Analog/Mixed-Signal IC Design Based on Number Theory",IEEE 14th International Conference on Solid-State and Integrated Circuit Technology, Qingdao, China (Nov. 2018)(IEEE Xplore)

[8] Richen Jiang, Gopal Adhikari, Yifei Sun, **Dan Yao**, Rino Takahashi,Yuki Ozawa, Nobukazu Tsukiji, Haruo Kobayashi, Ryoji Shiota, "Gray-code Input DAC Architecture for Clean Signal Generation",IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Xiamen, China (Nov. 6-9, 2017).IEEE Xplore

[9] **Dan Yao**, Yifei Sun, Masashi Higashino, Shaiful Nizam Mohyar, Tomonori Yanagida, Takuya Arafune, Nobukazu Tsukiji, Haruo Kobayashi, "DAC LINEARITY IMPROVEMENT WITH LAYOUT TECHNIQUE USING MAGIC AND LATIN SQUARES,"IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Xiamen, China (Nov. 6-9, 2017).IEEE Xplore

Domestic Conferences / Seminars

[1] Haruo Kobayashi, Kentaroh Katoh, Shuhei Yamamoto, Yujie Zhao, Shogo Katayama Jianglin Wei, Yonglun Yan, **Dan Yao**, Xueyan Bai, Anna Kuwana,"Waveform Sampling Technologies for Next Generation Communication and IoT Systems, "The 11th International Science, Social Sciences, Engineering and Energy Conference (I-SEEC 2022) The 6th International Conference on Technology and Social Science (ICTSS 2022), Kiryu, Japan (Online with Onsite Events), (December 25-28, 2022)

[2] Haruo Kobayashi, Xueyan Bai, Yujie Zhao, Shuhei Yamamoto, **Dan Yao**, Manato Hirai, Jianglin Wei, Shogo Katayama, Anna Kuwana,"Smart Mathematics Leads to Sophisticated Analog/Mixed-Signal Circuit", 5th International Conference on Technology and Social Science (ICTSS 2021) Kiryu, Japan, (Dec. 7-9, 2021) Online

[3] Zifei Xu, Xueyan Bai, **Dan Yao**, Anna Kuwana, Haruo Kobayashi,"Revival of Asynchronous SAR ADC based on Hopfield Network", 5th International Conference on Technology and Social Science (ICTSS 2021), Kiryu, Japan, (Dec. 7-9, 2021) Online

[4] **Dan Yao**, Anna Kuwana, Haruo KobayashiS,"EGMENTED-TYPE DAC UNIT CELL SELECTION ALGORITHM BASED ON EULER'S KNIGHT TOUR & MAGIC SQUARE", 2020 年度（第 11 回）電気学会東京支部栃木・群馬支所合同研究発表会オンライン開催（2021 年 3 月 1, 2 日）

[5] ケイ浩，桑名杏奈，白雪妍，**姚丹**，小林春夫（群馬大学）「CFD

技術を用いたブレード数の異なる S 字型風車の最適形状の検討」pp.76-79（21-25）2020 年度（第 11 回）電気学会東京支部栃木・群馬支所合同研究発表会,オンライン開催（2021 年 3 月 1, 2 日）

[6] **姚丹**，白雪研，桑名杏奈，小林春夫「単位セル 2 次元レイアウト配列のユナリ DA 変換器のダイナミックエレメントマッチング技術」2021 年度（第 12 回）電気学会東京支部 群馬支所・栃木支所 合同研究発表会，オンライン開催（2022 年 3 月 1 日, 2 日）

[7] 白雪妍，片山翔吾，**姚丹**，桑名杏奈，小林春夫,「小規模スイッチトキャパシタ回路を用いた非同期 SAR ADC」,2021 年度（第 12 回）電気学会東京支部 群馬支所・栃木支所 合同研究発表会，オンライン開催（2022 年 3 月 1 日, 2 日）

[8] Hao Xing, Anna Kuwana, Bai Xueyan, **Yao Dan**, Haruo Kobayashi, "Examination of Optimum Shape of Savonius Wind Turbine with Different Number of Blades using CFD Technology",4th International Conference on Technology and Social Science (ICTSS 2020), Kiryu, Japan, (Dec. 2-4, 2020)

[9] Xueyan Bai, **Dan Yao**, Anna Kuwana, Haruo Kobayashi,"Numerical Simulation for Optimization of Unsteady Rotating Wind Turbine"3rd International Conference on Technology and Social Science (ICTSS2019) , Kiryu, Japan (8-10 May, 2019)

[10] Qigong Teng, Xueyan Bai, **Dan Yao**, Anna Kuwana, Haruo

Kobayashi (Gunma Univ.),"Examination of Optimum Shape of 3-stage Savonius wind Turbine using CFD Technology", 5th Taiwan and Japan Conference on Circuits and Systems (TJCAS 2019 at Nikko), Nikko, Tochigi, Japan, August 19-21, 2019

[11] Bai Xueyan, **Yao Dan**, 桑名 杏奈, 小林 春夫「非定常回転風車の最適化のための数値シミュレーション」平成 30 年度　第 9 回 電気学会東京支部栃木・群馬支所 合同研究発表会,小山高専　2019 年 3 月 4 日(月), 5 日(火)

[12] P075 "Numerical Simulation for Characteristic Analysis of Vertical Axis Wind Turbine", **Dan Yao**, Anna Kuwana and Haruo Kobayashi Gunma University, Japan

[13] Yifei Sun, Shu Sasaki, **Dan Yao**, Nobukazu Tsukiji, Haruo Kobayashi,"Study on Digital Multiplier Architecture Using Square Law and Divide-Conquer Method",International Conference on Mechanical, Electrical and Medical Intelligent System 2017 (ICMEMIS2017), Nov. 29, 30 & Dec. 1, 2017 (Kiryu, Japan).

[14]佐々木秀, 孫逸菲, **姚丹**, 小林春夫,「2 乗則を用いたデジタル乗算器の研究」第 7 回 電気学会東京支部栃木・群馬支所 合同研究発表会, 於 足利工業大学 2017 年 3 月 2 日（木), 3 月 3 日（金）

[15]**姚丹**, 孫逸菲, 東野将史, 荒船拓也, 小林春夫,「ラテン方陣, 魔方陣レイアウトアルゴリズムを用いた DA 変換器線形性向上」　（レイ

アウトの重要性）第 7 回 電気学会東京支部栃木・群馬支所 合同研究発表会,於 足利工業大学 2017 年 3 月 2 日（木), 3 月 3 日（金）