SFDR 性能の DAC アーキテクチャでの比較

シャイフル　ニザム　ビン　モーヤ*,　　小林 春夫（群馬大学）

DAC Architecture Comparison for SFDR Performance

Shaiful Nizam Bin Mohyar*, Haruo Kobayashi (Gunma University),

(Keywords: SFDR, Current-steering DAC, Mismatch, Glitch, Fibonacci Sequence)

1. INTRODUCTION

Recently, emerging of wireless communication devices marketing such as smartphone, wireless modem and avionic parts has increases demands for high-resolution and high-speed digital-to-analog converters (DACs) with high spurious free dynamic range (SFDR) [1,2].

Fig. 1 shows a DAC uses in transmitter line to convert the digital values from digital signal processing (DSP) to analog values before filtering and amplification for transmission. DAC non-idealities are commonly sourced by current source mismatches and imperfect switch due to fabrication process variation. As a result, these two non-idealities will produce distortion signals at DAC output, and SFDR level is degraded. In this work, we are focusing on reducing the interference caused by these non-idealities.

There are several methods available such as Return-to-Zero (RZ) circuit for error correction design, differential-quad switching (DQS) and cascade transistors for switching techniques and digital data reshuffling. Here, we employ digital data reshuffling techniques which is easier to realize in digital circuit and CMOS implementation

Three different DAC architectures; binary, thermometer and Fibonacci sequence [3] are adopted. By aiming for linearity improvement to obtain better SFDR, their intrinsic redundancy, especially in Fibonacci sequence based DAC is exploited. Our MATLAB simulation shows their comparison results (in case that the static current source mismatches are considered).

This paper consists of seven sections. Section 2 describes current steering DAC architectures which are investigated. Section 3 discusses redundancy of each structure. Section 4 explains the nonlinearity output due to current source mismatches and glitch effects. Section 5 discusses Fibonacci sequences based current-steering DAC code selection for linearity improvement. The results and the conclusion are provided in Section 6 and Section 7.

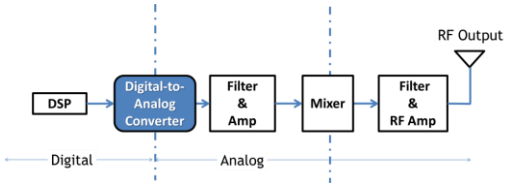

Fig. 1　A digital-to-analog converter in transmitter line.

2. CURRENT STEERING DAC ARCHITECTURE

A simple current-steering DAC uses binary weighted architecture (Fig.2 (a)), where current source values are binary-weighted ($I_1 = I$ $I_2 = 2I$ $I_3 = 4I$). When the digital input is 4, then SW3 turns on and SW1, SW2 turn off, and the current 4I ($=I_3$) flows into the resistor R and the output voltage, V_{out} of 4IR is produced. The binary weighted current-steering DAC has advantages of high speed sampling operation, low power and small chip area. However, large glitch energy, non-monotonicity input-output characteristics and no intrinsic redundancy are its drawbacks.

On the other hand, unary architecture of the current-steering DAC is introduced to overcome the binary-weighted disadvantages (Fig. 2(b)). This architecture offers low glitch power energy due to its identical weight of all 2^N-1 unit current sources that are used for N-bit resolution to obtain output optimization and flexibility of current source selection due its high intrinsic redundancy. In additional, the input-output monotonicity characteristics are guaranteed.

Ideally, all current cells are defined as in eq. (1).

$$I_1 = I_2 = I_3 = I_4 = I_5 = I_6 = I_7 = I \quad (1)$$

When the digital input is 4, then SW1, SW2, SW3, SW4 turn on and SW5, SW6, SW7 turn off, and the current 4I ($=I_1+I_2+I_3+I_4$) flows through the resistor R and the output voltage, V_{out} is 4IR. The drawbacks of this architecture are large chip area as well as certain amounts of power increase and sampling speed decrease.

In many cases, their combination is used; for higher bits, the segmented structure (Fig. 2(c)) is used while for lower bits the binary weighted structure is used. This

combined topology can achieve well-balance of,

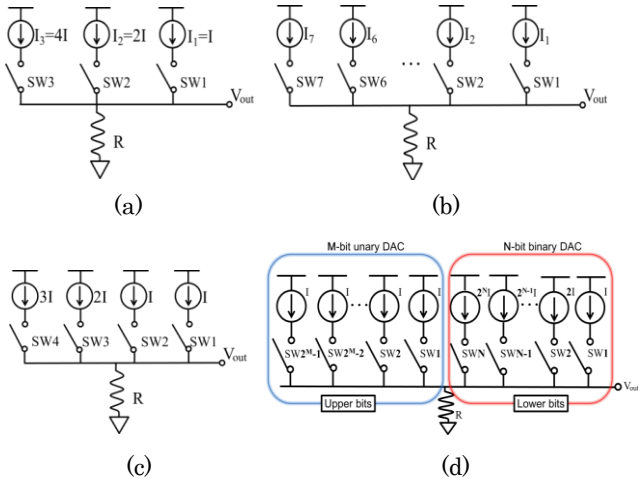

Fig. 2 A 3-bit current-steering DAC. (a) Binary weighted. (b) Unary. (c) Segmented. (d) Fibonacci sequence.

chip area [4], power, speed and glitch energy [5]. However, the segmented structure is eventually still containing genetic problem of unary architecture which becomes more complicated and more chip area for high resolution design.

In this paper, we investigate another DAC architecture which is based on Fibonacci sequence (Fig. 2(d)). Basically, this architecture is similar to binary weighted structure but uses different weighted values.

When the digital input is 4,

- SW1, SW2 and SW3 ➡ 4I(=I+I+2I)
- SW1 and SW4 ➡ 4I(=I+3I)
- SW2 and SW4 ➡ 4I(=I+3I)

Then, only one of above combination switches is turn ON and produce the current of 4I flows through the resistor R and the output voltage, V_{out} is 4IR. The advantages of this architecture are occupied small chip area compared to unary structure, high speed sampling and low power. However, this architecture also suffers from large glitch energy and non-monotonicity characteristics. Different with a binary weighted structure, Fibonacci sequence produces flexibility of current source selection which is better than binary weighted structure but less than unary structure, especially for glitch energy reduction to obtain output optimization.

3. REDUNDANCY

Fibonacci sequence can be expressed as follows:

$F_0 = 0,\ F_1 = 1,\ F_{n+2} = F_{n+1} + F_n,$ where $n \geq 0$; (2)

From eq. (2), generated Fibonacci number is 0, 1, 1, 2, 3, 5, 8, 13, 21,… where every n-th number is the summation of its previous two numbers. Table 1 shows that the availability of redundancy step increases as the number of bits (resolution) increases.

Table 1 Comparison of redundancy step and code combination

Comparison between Binary, Unary & Fibonacci code									
	Binary (B_n)			Unary (U_n)			Fibonacci (F_n)		
Bit, n	Weights, w_n	Step, n	Combination 2^n (max)	Weights, w_n	Step, 2^n-1	Combination $2^{2n}-1$ (max)	Weights, w_n	Step, k	Combination (max)
2	2,1	2	4 (1)	$1_3,1_2,1_1$	3	8 (3)	2,1,1	3	7 (2)
3	4,2,1	3	8 (1)	$1_7,\ldots,1_1$	7	128 (35)	3,2,1,1	4	16 (3)
4	8,4,2, 1	4	16 (1)	$1_{15},\ldots,1_1$	15	32768 (6435)	8,5,3, 2,1,1	6	51 (5)
5	16,8,4, 2,1	5	32 (1)	$1_{31},\ldots,1_1$	31	2.1×10^9 (3.0×10^8)	13,8,5, 3,2,1,1	7	126 (6)

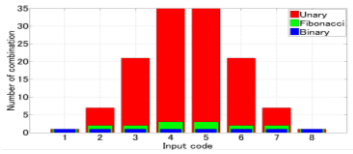

Fig. 3 Comparison of current source selection (3-bit).

For example, a 3-bit current-steering DAC of the binary, the unary and the Fibonacci structures uses 3, 4 and 7 current source cells respectively. Hence, Fibonacci structure consumes as smaller as the binary chip area compared to the unary structure. Fig. 3 shows the number of combination for each input code provided by these three architectures. Thus, these extra combination provide flexibility of current source selection for the distortion reduction caused by data-code dependent.

However, this architecture requires a flexible decoder to convert a binary code to a suitable Fibonacci code for switch controlling. As extended work, the decoding method of optimization current source selection for better distortion suppression is also investigated. Further explanation will be provided in section 5.

4. CURRENT-STEERING DAC NON-LINEARITY

Static non-linearity of the current-steering DAC is caused by current source mismatches, while dynamic performance SFDR, is degraded due to glitch effects as well as current source mismatches.

4.1 CURRENT SOURCE MISMATCH

The current source mismatches are inevitable inside an actual chip due to fabrication process variation and also current leakage. Ideally all currents I_1 - I_7 are the same in Fig. 2(b) as shown in eq. (1), however in reality there are current source mismatches and we define as follows:

$I = (I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7) / 7$ (3)

$I_k = I + dI_k \quad (k = 1, 2, .., 7)$ (4)

I is the average current, and dI_k is the deviation of the *k*-th current source I_k from the average current I. It follows from (3), (4) that

$dI_1 + dI_2 + dI_3 + dI_4 + dI_5 + dI_6 + dI_7 = 0.$ (5)

These current source mismatches cause the nonlinearity

of the current-steering DAC.

4.2 GLITCH

Glitch is caused when some current sources turn on and other current sources turn off simultaneously with timing mismatch for digital input change. This glitch effect is severe for the binary-weighted current steering DAC architecture (Fig. 4).

Fig. 4(a)-(d) illustrates an example of glitch effect during mid-code transition in a 4-bit binary-weighted DAC. B0-B3 are switches used to control the current flow of current sources with different weights. These examples demonstrate 0 to 15 possible digital inputs with 7 and 8 as a mid-code for Most Significant Bit (MSB).

Fig. 4(a) shows switch configuration in case that the digital input is 7 (0111). The switch configuration changes simultaneously as the digital input changes from 7 (0111) to 8 (1000). During this transition, the switch configuration has possibility to change either to 0 (0000) or 15 (1111). The transitions in Fig. 4(b) and Fig. 4(c) depend on how fast switch B3 is triggered to the desired input. Fig. 4(d) shows that desired input of 8 (1000) is obtained through transition of input 7 (0111). The produced glitch during transition will affect adversely dynamic performance. The glitch area in sampling period increases with the sampling frequency.

On the other hand, the unary architecture offers a good solution to reduce glitch [5]. For digital input increase, some current switches turn on and no current switches turn off, while for digital input decrease, some current switches turn off and no current switches turn on, and hence glitch energy is small. In case of Fibonacci structure, the flexibility of current source selection can be employed to reduce number of switches turn ON and OFF at each period which is not available in conventional binary structure.

5. CODE SELECTION

In this paper we use the conventional binary and unary code selection. In other hands, to observe the effectiveness of the Fibonacci structure current source flexibility, we use two different code selections. The changing of the code selection at every cycle as presented in [6] can reduces the effect of distortion caused by data-code dependent using data-weighted-averaging (DWA) method. The mismatches are averaged by time during conversion. Thus, the distortion power is distributed in frequency domain. Here, we propose the code selection for Fibonacci by using Look-Up Table (LUT). Fig. 5(a) and (b) shows the proposed Fibonacci sequence based DAC using counter and LUT based decoder.

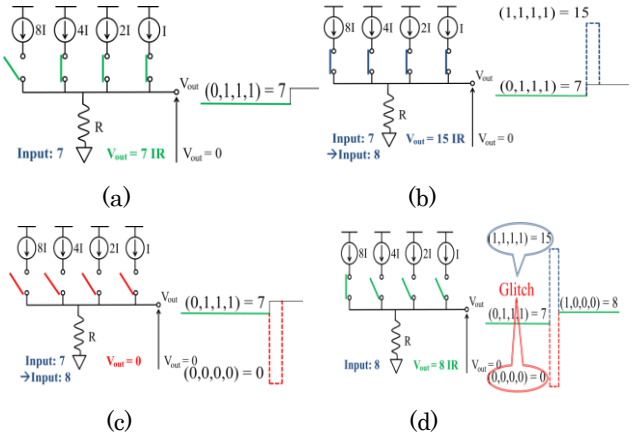

Fig. 4 Explanation of the glitch problem due to timing skew among switch control signals for the current sources in a binary-weighted current-steering DAC.

5.1 FIXED COUNTER

In Fig. 5(a), each input code is expressed by minimum of two combinations except input code of 7. This method selects the combination code of each digital input value by the address generated by a 2-bit counter. Hence, use of the same combination is prevented in the next same input value. By changing the variation of current source, the current source mismatches are cancelled out between each other as in eq. (5), leads to the reduction of the distortion caused by current source mismatches. Hence, output optimization can be obtained.

For example, with the input code of 7, 4, 1, 5, 5, 3.... Clock = 0, input code = 7, the DSP output is (b_2,b_1,b_0) = (1,1,1) with the counter (q_1,q_0) = (0,0). Then, the formed address thru the combination of DSP output and 2-bit counter is (b_2,b_1,b_0,q_1,q_0) = (1,1,1,0,0), which corresponds to (1,1,1,1) in the LUT. So, SW1 to SW4 turn ON. Next, clock = 1, input code = 4, the DSP output is (b_2,b_1,b_0) = (1,0,0) while the counter (q_1,q_0) = (0,1). The formed address is (b_2,b_1,b_0,q_1,q_0) = (1,0,0,0,1), and output code is (1,0,0,1). This time, SW1 and SW4 remain while SW2 and SW3 turn OFF. When clock = 2, input code = 1, in, the DSP output is (b_2,b_1,b_0) = (0,0,1) while the counter (q_1,q_0) = (1,0). Then, the address becomes (b_2,b_1,b_0,q_1,q_0) = (0,0,1,1,0) and output code is (0,0,0,1). Next, SW1 remains while SW4 turns OFF. So, this procedure will continue until the end of input code.

5.2 RANDOM COUNTER

This method can use a random value generated by 2-bit randomizer. In this case, the random value is from 00 to 11. With the same input code as in method 1, this method

procedure as follows:

Clock = 0, input code = 7, the DSP output is (b_2,b_1,b_0) = (1,1,1) while the randomizer (q_1,q_0) = (1,0). By randomizing, the address becomes (b_2,b_1,b_0,q_1,q_0) = (1,1,1,1,0), equal to (1,1,1,1) in the LUT. So, SW1 to SW4 turn ON. Next, clock = 1, input code = 4, the DSP output is (b_2,b_1,b_0) = (1,0,0) while the randomizer (q_1,q_0) = (0,0). The address is (b_2,b_1,b_0,q_1,q_0) = (1,0,0,0,0), and output code is (0,1,1,1). This time, SW1 and SW4 remain ON while SW2 and SW3 turn OFF. When clock = 2, input code = 1, the DSP output is (b_2,b_1,b_0) = (0,0,1) while the randomizer (q_1,q_0) = (1,1). Then, the address is (b_2,b_1,b_0,q_1,q_0) = (0,0,1,1,1), output code is (0,0,1,0). Next, SW1 remains while SW4 turns OFF. So, this procedure will continue until the end of input code.

6. SIMULATION RESULT

Verification of this work has been performed using MATLAB under condition as summarized in Table 2. Simulation results show that the Fibonacci structure obtained a comparable SFDR compared to the conventional binary structure but less than unary structure. Fig. 6(a) shows the ideal SFDR. In case of mismatches presented, Fig. 6(b)-(e) show the SFDR performances with the mismatch of 10% using unary, binary and Fibonacci with both fixed and random counter selections (refer Table 3).

Table 2 Simulation conditions

Input signal		Error	
Input frequency, fin	0.2 – 1GHz	Number of errors	10 - 1023
Sampling frequency, fs	2.048 GHz	Mismatch (%)	10
FFT point	32768		

7. CONCLUSION

This paper has investigated the SFDR performance of three different DAC architectures by employing their intrinsic redundancy. We expect that Fibonacci structure can produces better SFDR performance compared to binary structure due its intrinsic redundancy but less than unary structure. Fibonacci structure DAC can be implemented simply compared to unary and the Fibonacci DAC compromises power, area and SFDR performance.

Table 3 Summary of SFDR performance (10-bit)

Architecture	Algorithm	SFDR (dBc)	Diff (dB)
Ideal	-	83.4	-
Unary	Thermometer coded (TC)	80.2	- 3.2
Binary	-	76.9	- 6.5
Fibonacci	Counter (I)	75.7	- 7.7
	Random (II)	75.7	- 7.7

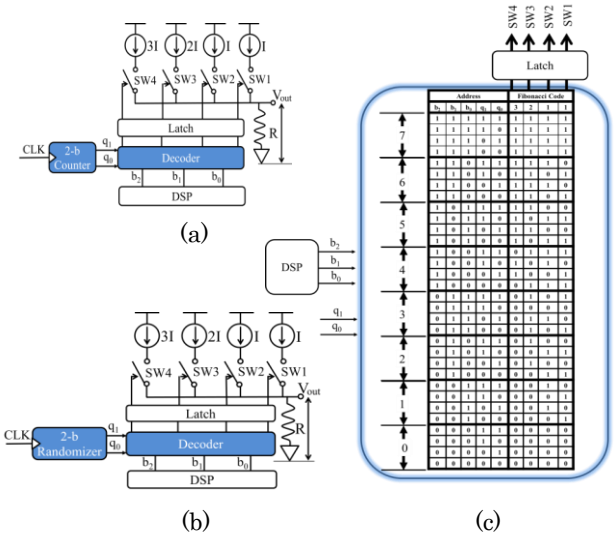

Fig. 5 Fibonacci sequence based current-steering DAC using different selection methods. (a) Counter. (b) Randomizer. (c) Look Up Table (LUT) based decoder.

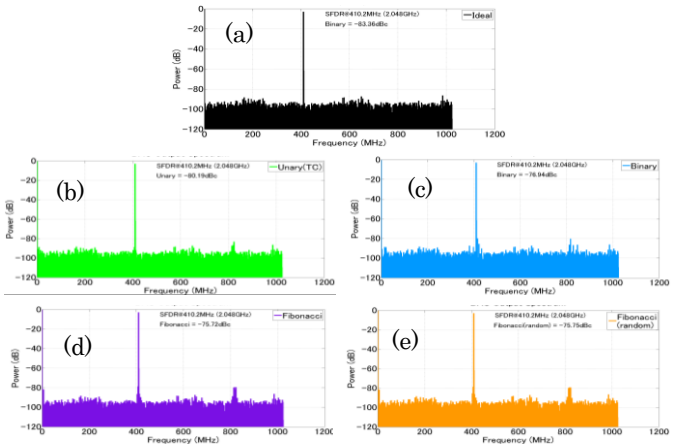

Fig. 6 SFDR performance. (a) Ideal case. (b) Unary. (c) Binary. (d) Fibonacci (fixed counter selection) (e) Fibonacci (random counter selection).

REFERENCES

[1] R. J. van de Plassche, CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters, Springer, 2010.

[2] R. J. van de Plassche, Integrated Analog-to-Digital and Digital-to-Analog Converters, Kluwer Academic Publishers, 1994.

[3] Y. Kobayashi, M. Kazumi, Y. Zhixiang, H. Kobayashi, "ADC/DAC Redundancy Design Using Fibonacci Sequence",

[4] C. H. Lin and K. Bult, "A10-b, 500-MSample/s CMOS DAC in 0.6 mm^2", IEEE Journal of Solid-State Circuits, vol.33, no.12, pp.1948-1958, December 1998.

[5] A. Van den Bosch, M. Borremans, J. Vandenbussche, G. Van der Plas, A. Marques, J. Bastos, M. Steyaert, G. Gielen, W. Sansen, "A 12-bit 200 MHz Low Glitch CMOS D/A Converter", IEEE Custom Integrated Circuits Conference, Santa Clara, CA, pp.249-252, May 1998

[6] H. San, H. Kobayashi, S. Kawakami, N. Kuroiwa, "A Noise-Shaping Algorithm of Multi-bit DAC Nonlinearities in Complex Bandpass Δ ΣAD Modulators", IEICE Trans. on Fundamentals, E87-A, no.4, pp.792-800, April 2004.