Gray-code Input DAC Architecture for Clean Signal Generation

Richen Jiang, Gopal Adhikari, Yifei Sun, Dan Yao, Rino Takahashi,
Yuki Ozawa, Nobukazu Tsukiji, Haruo Kobayashi, Ryoji Shiota †*
Division of Electronics and Informatics, Gunma University, 376-8515 Japan, *e-mail: koba@gunma-u.ac.jp
† Socionext Inc., Nomura Shin-Yokohama Bldg., 2-10-23 Shin-Yokohama, Kohoku-ku, Yokohama, 222-0033, Japan

Abstract- This paper describes digital-to-analog converter (DAC) architectures for clean signal generation used in such as arbitrary waveform generators (AWGs) as well as graphic displays; the clean signal is applied to analog/mixed-signal/RF devices for testing. We propose here three types of Gray-code input DAC architectures (current-steering, charge-mode and voltage-mode DACs) for glitch reduction. They use current/voltage switch matrixes realized by double-pole double-throw (dpdt) switches. For a binary-weighted DAC, glitches which are trigged by switching during conversion in a DAC system are serious problems in some applications. Gray code (reflected binary code) is a binary numeral system where two successive values differ in only one bit and conversion between binary-code and Gray-code can be easily realized by XOR circuits. SPICE simulation has been performed to prove their feasibility and ability of glitch reduction.

Index Terms —DAC, Gray Code, Glitch, AWG, Graphics Display, Signal Generation

I. INTRODUCTION

A digital-to-analog converter (DAC) is a function that converts digital data into analog signal, and the DAC is used for automatic test equipment (ATE) systems such as arbitrary waveform generators (AWGs) [1-3] as well as graphic displays. There clean DAC output is required for high frequency signal generation. This paper investigates DAC architectures for clean analog output by reducing glitches with Gray-code input topologies. Many analog/mixed-signal circuit designers have considered that coming up with Gray-code input DAC topologies is difficult; only a few Gray-code input DAC topologies have been proposed so far [4, 5, 11], but they may not be very systematic or their circuit topologies may be limited; References [4,5] do not discuss the extension to high resolution DACs and reference [11] does not mention about the charge-mode DAC. In this paper we propose three types of Gray-code input DAC topologies which are systematic and regular as well as expected to be efficient and practical.

The DAC can be realized with current sources, resisters and/or capacitors. DAC architectures can be divided as follows: (i) Unary-weighted DACs with current sources, resistances or capacitors which have the same value, thus they perform a thermal code to represent the input digital data. (ii) Binary-weighted DACs with binary valued sources, resistances or capacitors which are compatible with the binary digital data and help decrease the units used in a DAC. And the R-2R ladder DAC shows its ability of D/A conversion to be the same with the unary R-DAC while much resister area been released. (iii) Hybrid DACs that combine different kinds of DACs [6-8].

When it comes to the binary DAC, a disadvantage cannot be ignored in many applications, which is the glitch affecting the performance of D/A conversion. Assume that changing a binary code from a value to its successive value, for example, a binary code is changing between 011 ($3_{decimal}$) and 100 ($4_{decimal}$). Then all of three bits have to be reversed. In applications, DACs often work at a value near the middle point, in which a number of switching can lead to noise and performance degradation.

Glitch performance is important in applications such as DACs in a display driver. Switching of the MSB (most significant bit, high-order bit) can cause the most glitch (worst-case) (Fig.1, Table I) [7]. For example, when input data converting from 0111 to 1000, the switching-skew may experience a transient state of 0000 instead of directly change from 0111 to 1000. As a result, the conversion output shows a deep spike. This glitch effect is a serious problem for some applications such as graphic display (Fig. 2) as well as some analog /mixed-signal/RF device testing.

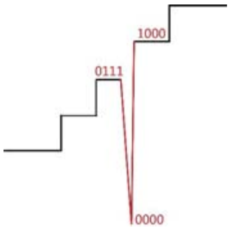
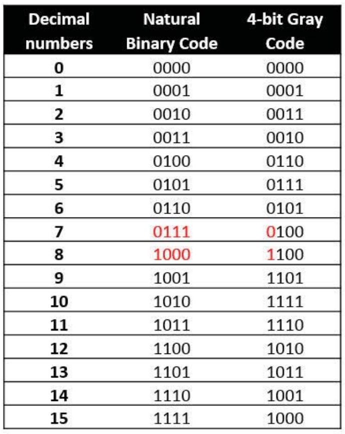


Fig. 1. Glitch caused by MSB change

Table I. Comparison of binary-code and Gray-code



Decimal numbers	Natural Binary Code	4-bit Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

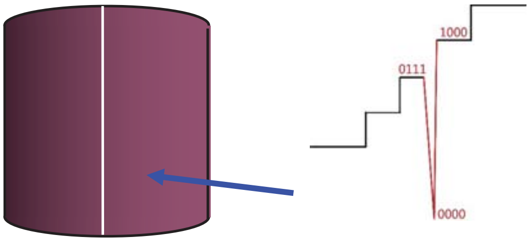


Fig. 2 DAC glitch effects to graphic display.

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

Gray-code (reflected binary code) is a binary numeral system where two successive values differ in only one bit (Fig. 2). So a Gray-code driven DAC is capable to reduce the glitch. The conversion between binary-code and Gray-code can be easily realized by XOR logic as followed (in 4-bit case):

Binary code: B3, B2, B1, B0
Gray code: G3, G2, G1, G0
$G2=B3\oplus B2$, $G1=B2\oplus B1$, $G0=B1\oplus B0$
$B3=G3$, $B2=G3\oplus G2$,$B1=G3\oplus G2\oplus G1$,$B0=G3\oplus G2\oplus G1\oplus G0$

While Gray-code utilized in ADCs and TDCs (time-to-digital converters) have been reported recently [9, 10]. The Gray-code input DAC is long considered to be unable to realize. This paper will present three kinds of Gray-code input DACs to show that the Gray-code input DACs are possible to construct systematically and they are able to reduce the glitch.

II. Gray-code input DAC architecture

A. Current/voltage switch matrix

In order to construct the Gray-code DAC, first we introduce the current/voltage switch matrix as a key component to bridge between binary-code arrayed current sources/capacitors/voltage sources and Gray-code input. The current/voltage switch matrix is actually a kind of double-pole double-throw (dpdt) switch [5, 6], which has two input terminal (In1, In2), two output terminal (Out1, Out2) and a control terminal (S). For each input terminal can only be connected with one output terminal, there only exists two kinds of connections, which are called parallel connection and cross connection. We define that when S=0, the dpdt is in parallel connection state and when S=1 is in cross connection state.

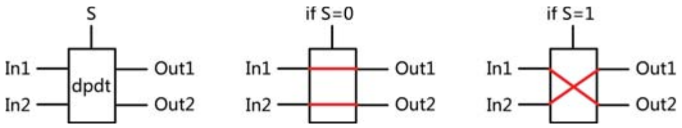


Fig. 3. Definition of voltage/current switch matrix

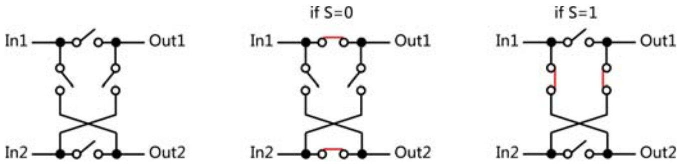


Fig. 4. Realization of voltage/current switch matrix by dpdt

B. Current-steering Gray-code DAC

First the current-steering Gray-code DAC is introduced. Fig.5 shows the conventional binary-weighted current-steering DAC. While Gray-code input current-steering DAC (Fig.6) replaces the switch array with dpdt array and sets one more binary-distributed current sources array. As for the output, the subtraction of I_{out-} and I_{out+} called I_{out} is defined as the output.

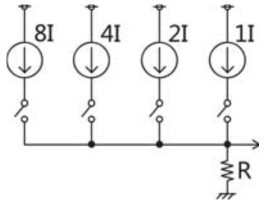


Fig. 5. Binary-weighted current-steering DAC

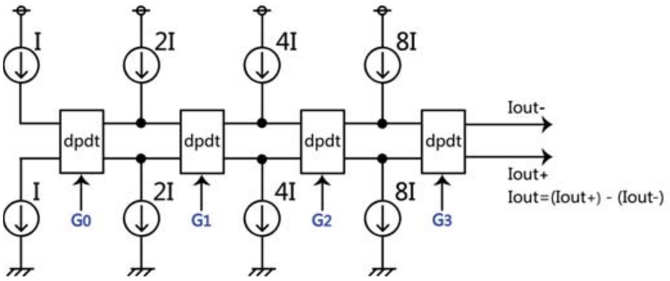


Fig. 6. Gray-code input current-steering DAC

The concept of the Gray-code input current-steering DAC is that the first input binary-code are converted to Gray-code through XOR logic, and Gray-code is in charge of the steering of dpdt to guide the binary-code weighted current sources. Thus Gray-code only controls the switching of current sources while the rest parts are compatible to conventional binary-code DAC architecture.

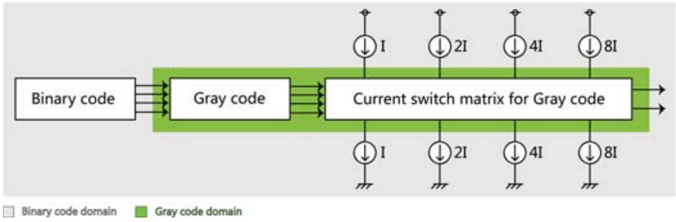


Fig. 7. Code domain in Gray-code input current-mode DAC

Fig. 8 shows an example of how the Gray-code input current-steering DAC operation when the input data=5. Three of the dpdt switches are set to be in cross connection state and one dpdt switch is in parallel connection state. Thus I_{out-}=(2I+8I)-(1I+4I)=5I, I_{out-}=(1I+4I)-(2I+8I)=-5I and I_{out}=(I_{out+})-(I_{out-})=-10I.

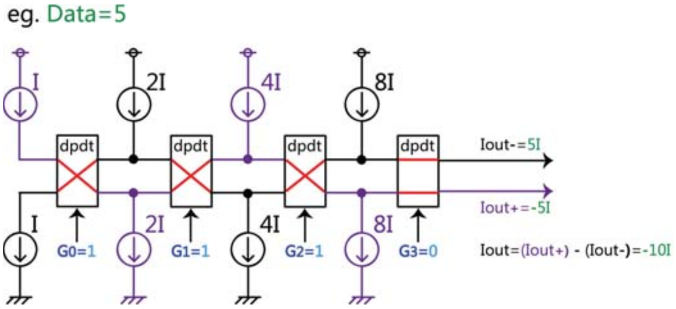


Fig. 8. A Gray-code input current-mode DAC (in case data=5).

C. Charge-mode Gray-code DAC

Next the Gray-code input charge-mode DAC has been designed by simply replacing binary-code controlled switches to Gray-code controlled dpdt switches (Fig.9, Fig.10). The rest parts remain the same and the Gray-code input charge-mode DAC also operates in a way of two-steps: sample mode where necessary capacitors are connected to the reference voltage source, and output mode where the output is obtained through an integrator amplifier circuit and all the capacitors are discharged at the same time [7]. Fig.11 and Fig.12 show the operation of the Gray-code input charge-mode DAC when the input data=5.

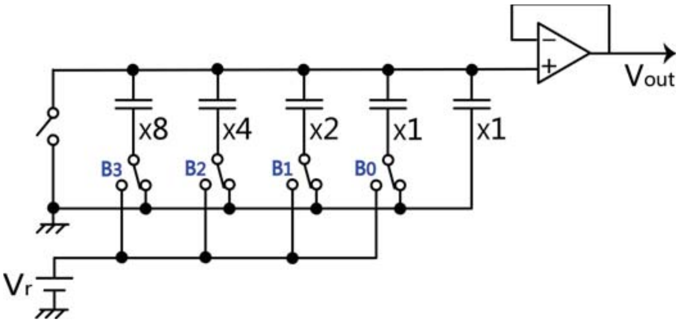


Fig. 9. A binary-weighted capacitor DAC

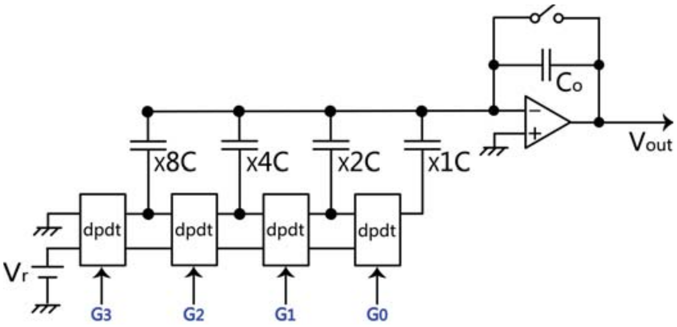


Fig. 10. A Gray-code input charge-mode DAC

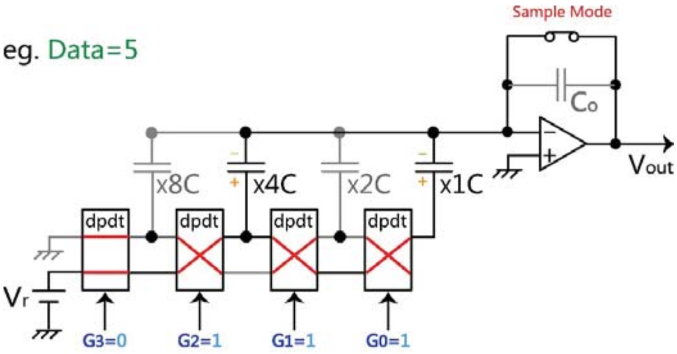


Fig. 11. Sample mode of a Gray-code input charge-mode DAC (data=5)

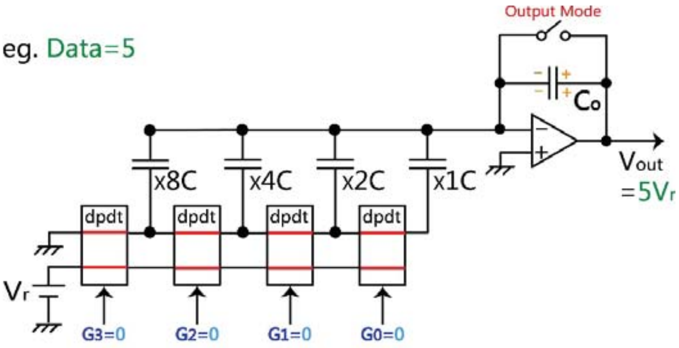


Fig. 12. Output mode of a Gray-code input charge-mode DAC (data=5)

D. Voltage-mode Gray-code DAC

The third kind of the Gray-code input DAC is called voltage-mode Gray-code DAC (Fig. 13). This time the original input Vr is set in series connection with dpdt switches and added with binary-code obeyed voltage sources by addition amplifiers. Fig.14 shows the operation of Gray-code input charge-mode DAC when the input data=5.

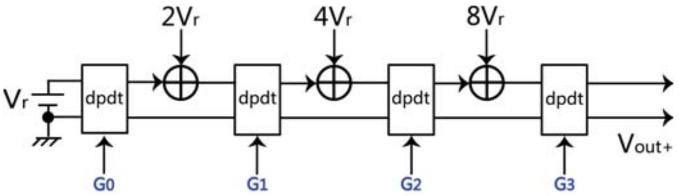


Fig. 13. A Gray-code input voltage-mode DAC

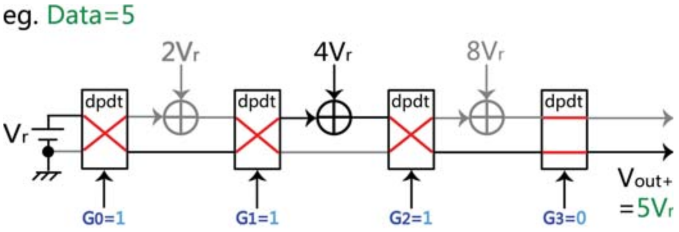


Fig. 14. A Gray-code input voltage-mode DAC (in case data=5)

III. SPICE Simulation of Gray-code DAC

In order to test and verify these three kinds of Gray-code input DACs and to see if their glitches can be reduced compared to the binary-code ones, we realized them in the environment of LTspice IV, which is a SPICE simulator produced by Linear Technology Corp.

A. Simulation of Current-steering Gray-code DAC

First we implemented the current-steering Gray-code DAC (Fig.15) which starts with a synchronous binary-code initiator. Then the binary-code converted to Gray-code by XOR logic and stored at the latch array. After D/A conversion, V_{out} can be obtained from I_{out-} and I_{out+} by passing through two output resistances, a subtraction amplifier and an S&H (sampling and holding) circuit.

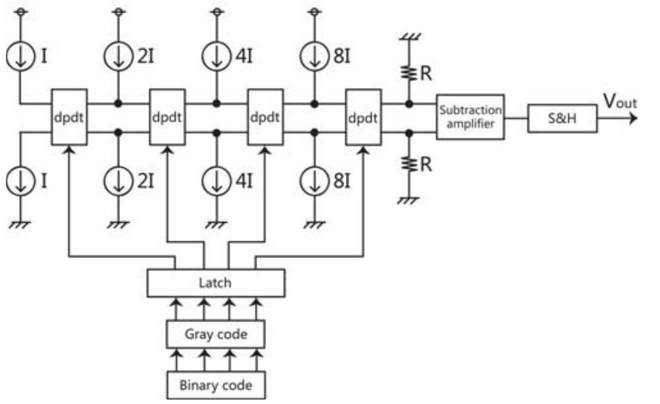


Fig. 15. Implementation of a Gray-code input current-mode DAC

An 8-bit current-steering Gray-code DAC simulation is performed with the following parameters:

Current sources: I_0=0.1mA, I_1=0.2mA, I_2=0.4mA, I_3=0.8mA, I_4=1.6mA, I_5=3.2mA, I_6=6.4mA, I_7=12.8mA;
Clock cycle: 2ms;
Digital input range: 0~1;
Analog output range: -25.5～25.5V.

The result is shown at Fig.16 and we can see the 00000000 ($0_{decimal}$) ~ 10000000 ($255_{decimal}$) digital input is converted to 0V~25.5V analog output correctly.

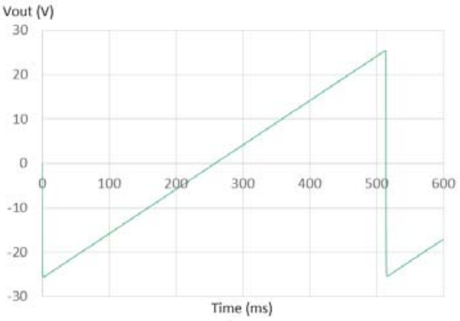


Fig. 16. Simulation result of Gray-code input current-mode DAC

B. *Simulation of Charge-mode Gray-code DAC*

Then we implemented the charge-mode Gray-code DAC (Fig.17). Here a non-overlap clock is added to perform the conversion of sample-mode and output-mode.

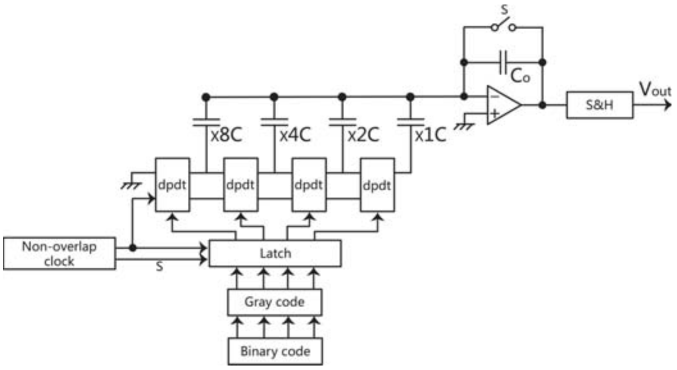


Fig. 17. Implementation of a Gray-code input C-DAC

An 8-bit charge-mode Gray-code DAC simulation is performed with the following parameters:

Capacitors: C_3=80pf, C_2=40pf, C_1=20pf, C_0=10pf, C_o=10pf;
Clock cycle: 2ms;
Digital input range: 0~1;
Analog output range: 0～25.5V.

The result is shown in Fig.18 and we can see that the 00000000 ($0_{decimal}$) ~ 10000000 ($255_{decimal}$) digital input is converted to 0V~25.5V analog output correctly.

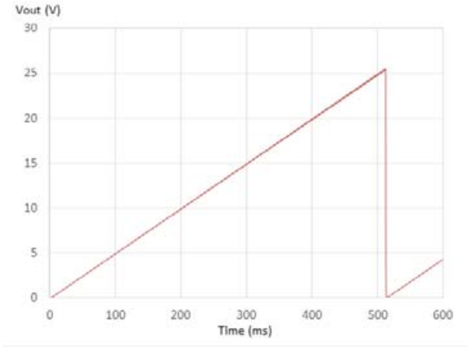


Fig. 18. Simulation result of a Gray-code input charge-mode DAC (in 8-bit case)

C. *Simulation of Voltage-mode Gray-code DAC*

Next we implemented the voltage-mode Gray-code DAC (Fig. 19).

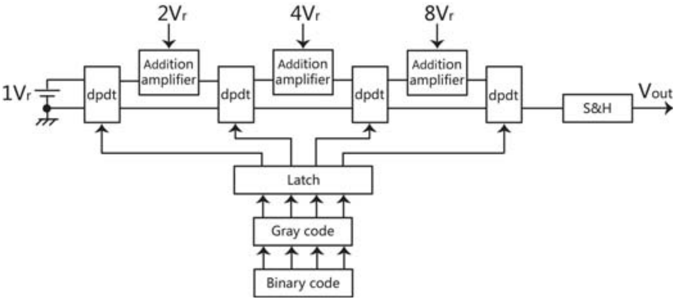


Fig. 19. Implementation of a Gray-code input voltage-mode DAC

An 8-bit voltage-mode Gray-code DAC simulation is performed with the following parameters:

Adder voltage at addition amplifiers: V_1=0.1V, V_2=0.2V, V_3=0.4V, V_4=0.8V, V_5=1.6V, V_6=3.2V, V_7=6.4V, V_8=12.8V;
Clock cycle: 2ms;
Digital input range: 0~1;
Analog output range: 0～25.5V.

The result is shown at Fig. 20 and we can see the 00000000 ($0_{decimal}$) ~ 10000000 ($255_{decimal}$) digital input is converted to 0V~25.5V analog output correctly.

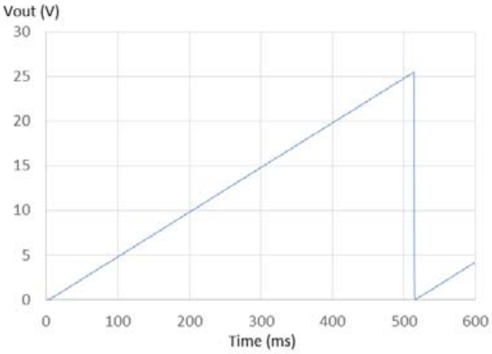


Fig. 20. Simulation result of a Gray-code input voltage-mode DAC (in 8-bit case)

D. Analysis of glitch reduction

In order to verify the glitch reduction performance of Gray-code input DAC, a conventional binary-weighted current-steering DAC with the same specification was simulated together with the Gray-code current steering DAC (the output of binary-weighted current-steering DAC is shifted downside by 25.5V to have a better comparison). A step-up delay (t_d=0.1ms) is set at the clock of each data latch to simulate the switching delay accumulation from LSB to MSB.The result is shown at Fig.21 and Fig.22. Binary-weighted current-mode DAC shows glitch distribution with the most significant glitch at the middle of the waveform when MSB changes its value. While Gray-code I-DAC keeps its waveform same to identical situation and sustain its monotony. Then we changed the delay to be random and unrelated in order (0ms~1ms) and simulated these two I-DACs (Fig.23). Gray-code I-DAC also shows its ability of glitch reduction.

IV. Conclusion

This paper presents three kinds of Gray-code input DACs (current-steering, charge-mode and the voltage-mode) to show that it is possible to steering the switches of DACs by Gray-code. Comparative SPICE simulations have been performed to verify that Gray-code input make contribution to glitch reduction. The proposed DAC architectures are expected to apply ATE systems and graphics displays for clean signal generation.

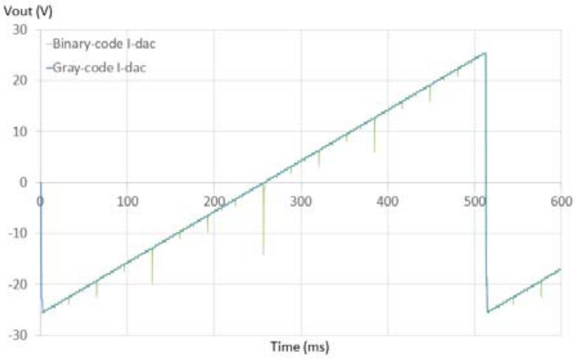


Fig. 21. Glitch comparison of binary-code and Gray-code input current-mode DACs (step-up delay, sweep up)

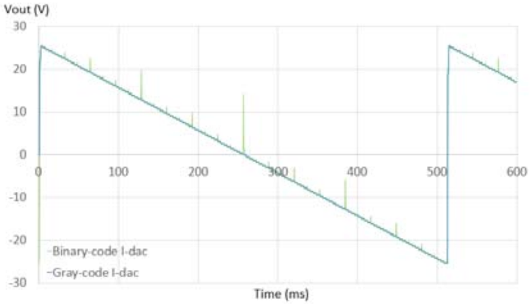


Fig. 22. Glitch comparison of binary-code and Gray-code input current-mode DACs (step-up delay, sweep down)

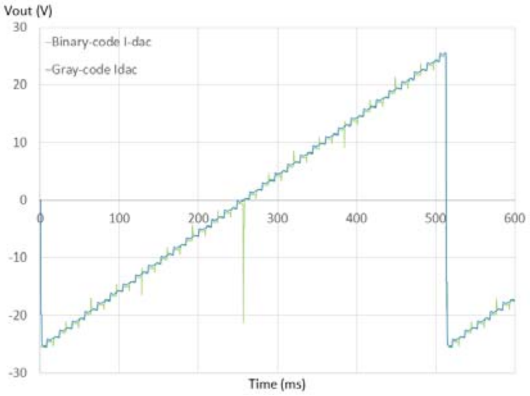


Fig. 23. Glitch comparison of binary-code and Gray-code input current-mode DACs (random delay, sweep up)

References

[1] Gordon Roberts, Friedrich Taenzler, Mark Burns, An Introduction to Mixed-Signal IC Test and Measurement, 2nd-edition, Oxford University Press (2011).

[2] Akinori Maeda, “A Method to Generate a Very Low Distortion High Frequency Sine Wave Using an AWG,” IEEE International Test Conference, Santa Clara, CA (2008).

[3] Fumitaka Abe, Yutaro Kobayashi, Kenji Sawada, Keisuke Kato, Osamu Kobayashi, Haruo Kobayashi, “Low-Distortion Signal Generation for ADC Testing,” IEEE International Test Conference, Seattle, WA (Oct. 2014).

[4] N. C. Seiler: “Gray Code DAC Ladder”, US Patent: 4591826 (May 1986)

[5] Simon Voigt Nesboe, Xingguo Xiong: “Gray-code Digital-to-Analog Converters (DACs) for Glitch Reduction”, Zone 1 Conference of the American Society for Engineering Education, Bridgeport, CT (April 2014)

[6] Franco Maloberti: Data Converters, Springer, (2007)

[7] Rudy van de Plassche: CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters (2nd Edition), Kluwer Academic Publishers, (2003).

[8] Phillip E. Allen, Douglas R. Holberg: CMOS Analog Circuit Design, (2nd-Edition), Oxford University Press, (2002)

[9] Haruo Kobayashi, Toshiya Mizuta, Kenji Uchida, et. al.: “A High-Speed 6-Bit ADC Using SiGe HBT”, IEICE Trans. Fundamentals, (Mar. 1998)

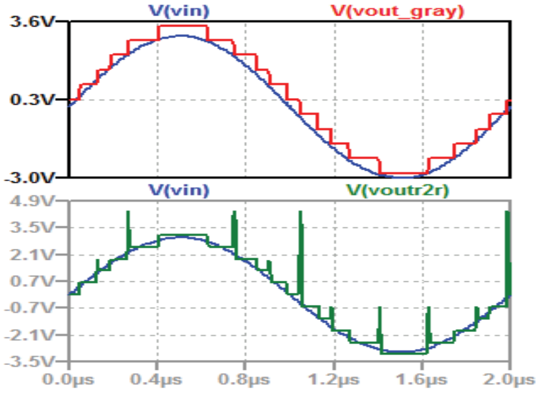
[10] Conbing Li, Haruo Kobayashi: “A Glitch-Free Time-to-Digital Converter Architecture Based on Gray Coode”, IEEE International Symposium on Radio-Frequency Integration Technology, Sendai, Japan (Aug. 2015).

[11] Gopal Adhikari, Richen Jiang, Haruo Kobayashi, "Study of Gray Code Input DAC Using MOSFETs for Glitch Reduction", IEEE 13th International Conference on Solid-State and Integrated Circuit Technology, Hangzhou, China (Oct, 2016).

APPENDIX

We have performed simulations for binary code input DAC and Gray code input DAC in 4-bit case, also using sinusoidal input signal, and compared their glitch performance.

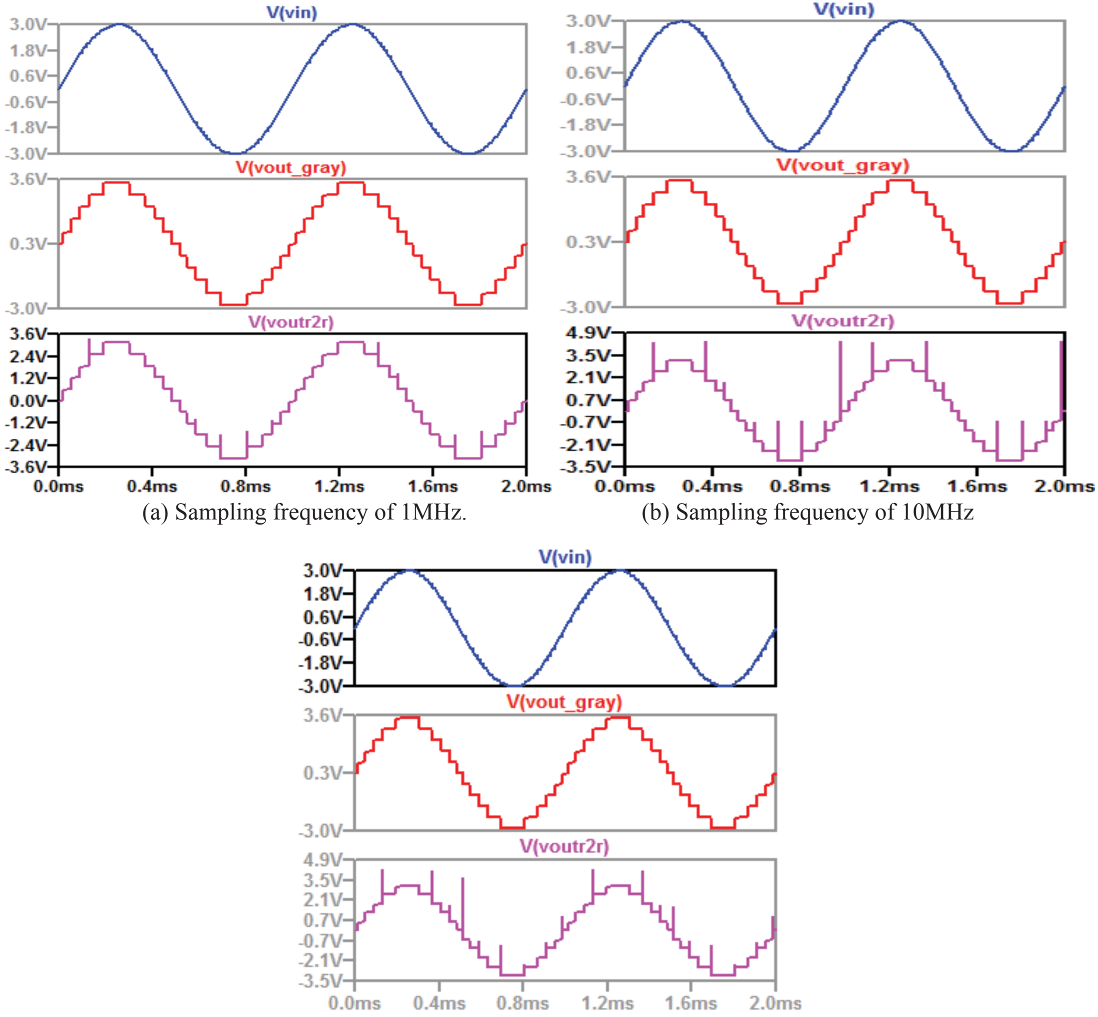
We see that the Gray code input DAC has better glitch performance in Fig. A1 and Fig. A2.



(a) Input frequency of 500kHz.

(b) Input frequency of 800kHz

Fig. A1: Sampling frequency of 50MHz.
Blue: Ideal DAC output. Red: Gray code input DAC output. Green: Binary code input DAC output.



(a) Sampling frequency of 1MHz.

(b) Sampling frequency of 10MHz

(c) Sampling frequency of 50MHz.

Fig. A2: Input frequency of 10kHz.
Blue: Ideal DAC output. Red: Gray code input DAC output. Magenta: Binary code input DAC output.
Toggle rate: Gray code 1/2 compared to binary code