ICSICT-2018
2018 IEEE 14th International Conference on Solid-State and Integrated Circuit Technology

S38-2 Analog Circuits Room H
Nov. 2 , 2018

DWA Algorithm for Band-Pass ΔΣ DAC with Ternary Unit Cells

Jun-ya Kojima   Nene Kushita   Masahiro Murakami
Anna Kuwana   Haruo Kobayashi

Gunma University, Japan

群馬大学
GUNMA UNIVERSITY

Outline

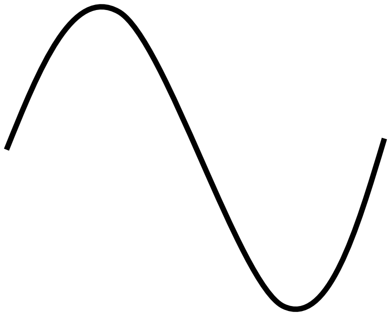
- Research Background
- ΔΣDA Converter
  - DWA*Algorithm (* Data-Weighted Averaging)
- Simulation Verification
  - Binary, Ternary DWA Overview
  - ΔΣ DA Converter : HP type
  - ΔΣ DA Converter : BP type
- Conclusion

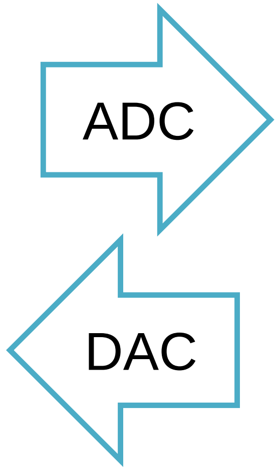
Outline

- Research Background
- ΔΣDA Converter
  - DWA*Algorithm (* Data-Weighted Averaging)
- Simulation Verification
  - Binary, Ternary DWA Overview
  - ΔΣ DA Converter : HP type
  - ΔΣ DA Converter : BP type
- Conclusion

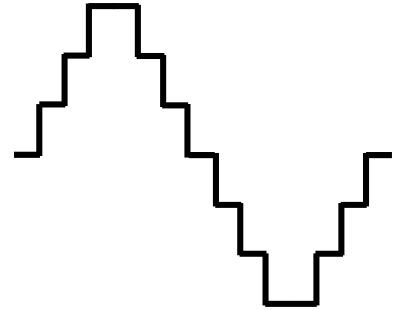
Research Background

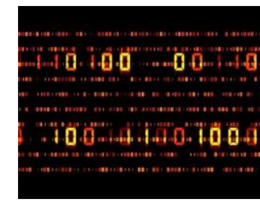
Analog signal





Digital signal



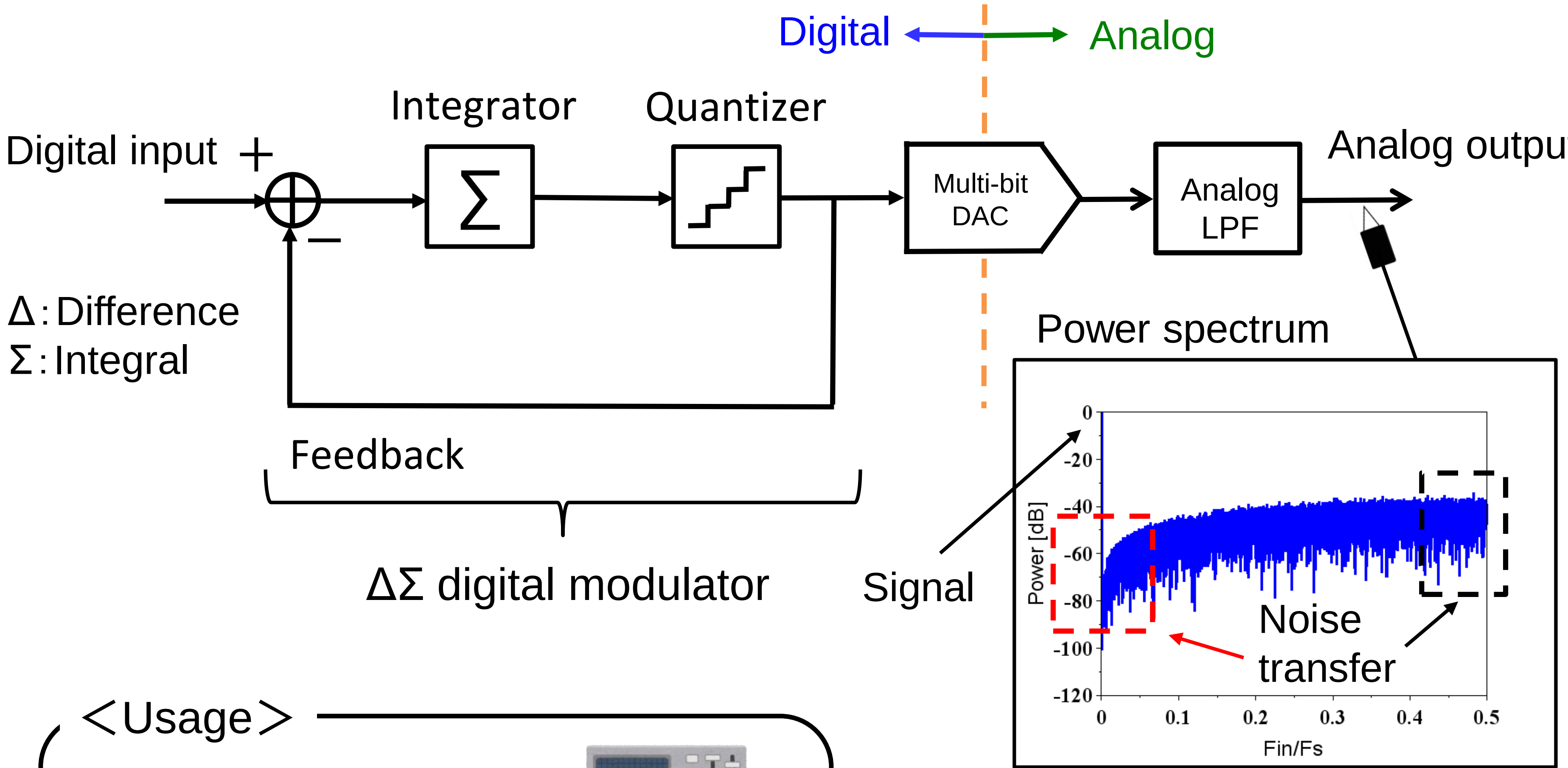


Continuous signal
Physical quantity existing in nature

Discrete signal
Binary number

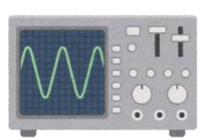
- ΔΣ Digital-to-Analog Converter (ΔΣDAC) →Required
  - Mostly digital circuit
  - High-resolution ,High-linearity
  - DC signal , low frequency signal generation

ΔΣ Digital to Analog Converter (Low Pass)



<Usage>

- Electric measurement

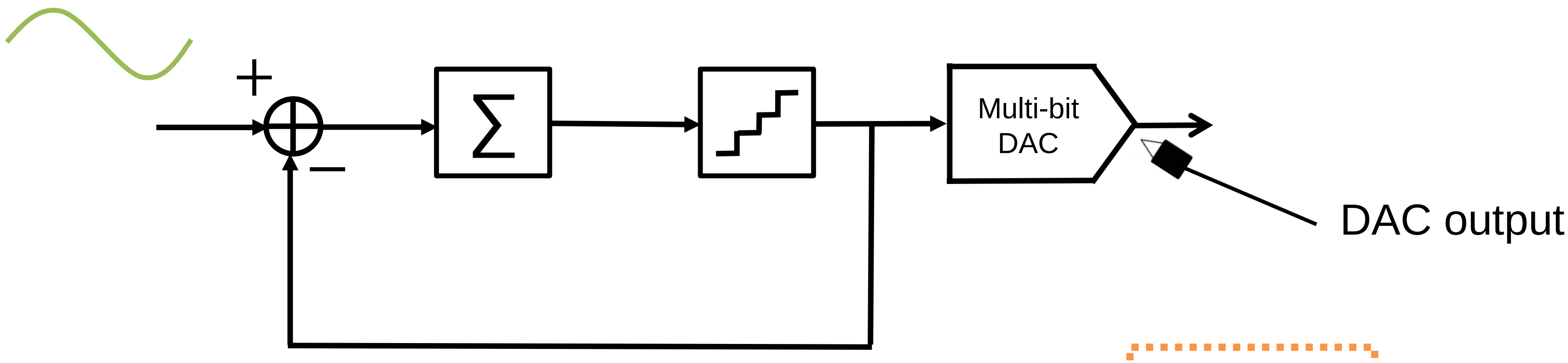


- Audio system

Noise

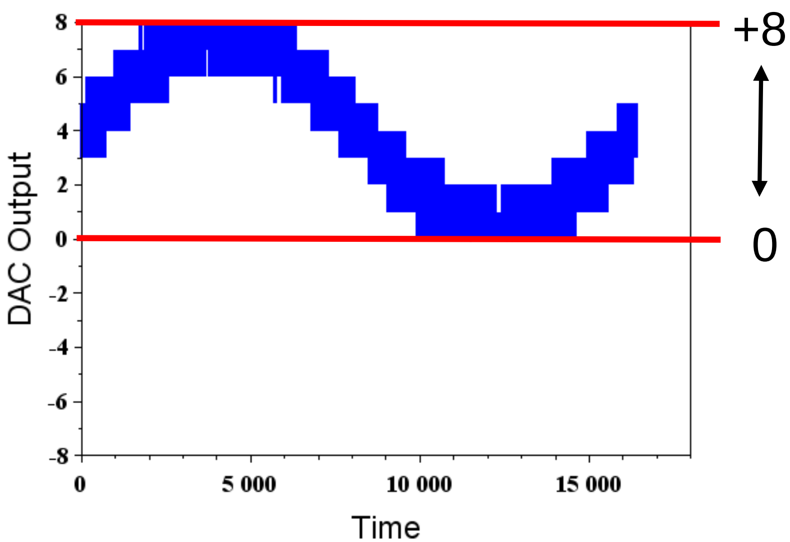
decrease at low frequency

Introduction to Ternary

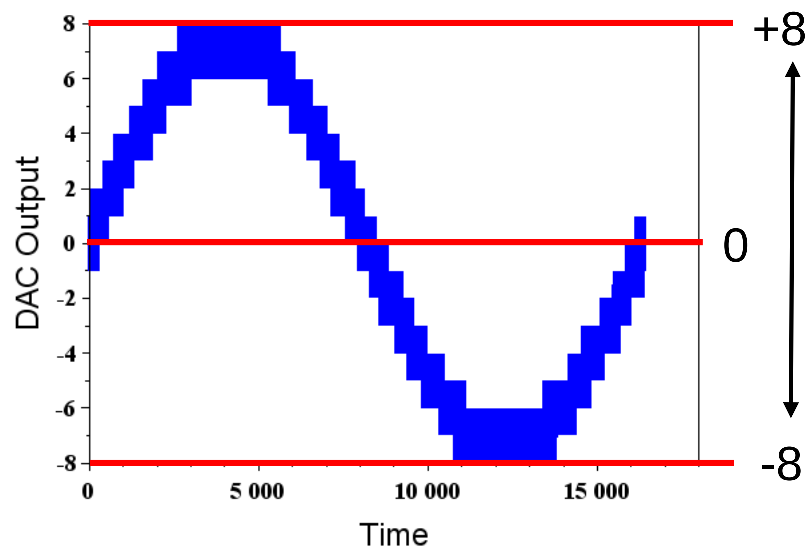


Forces

Binary ⇒ + and 0 value



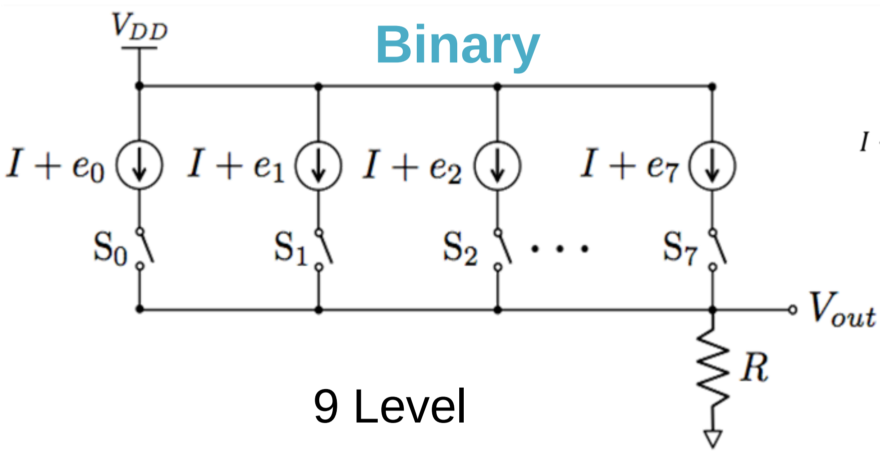
Ternary ⇒ +, - and 0 value

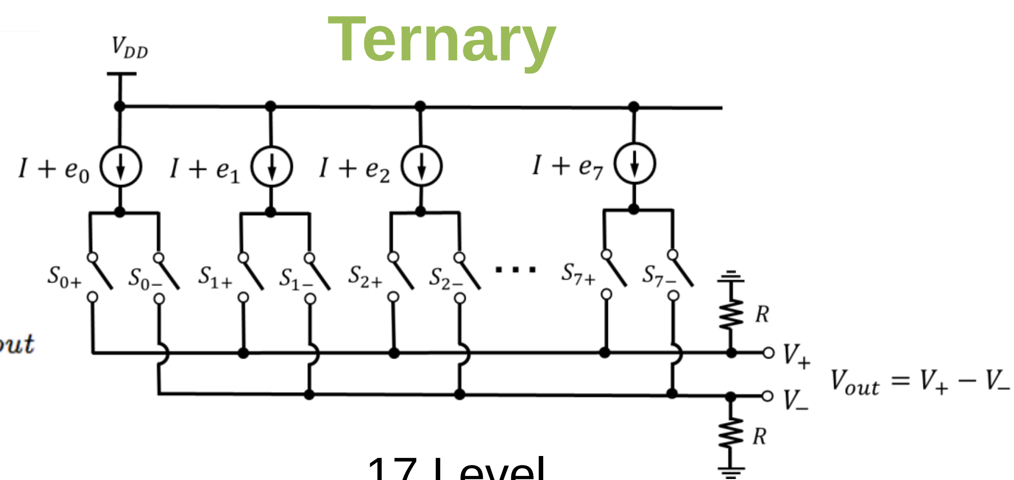


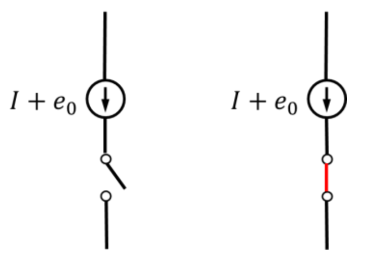
Reasons for Ternary Usage

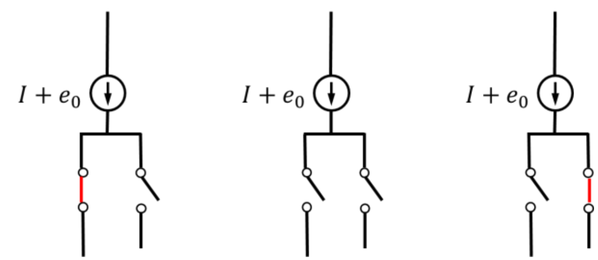
- Comparison

8 current sources









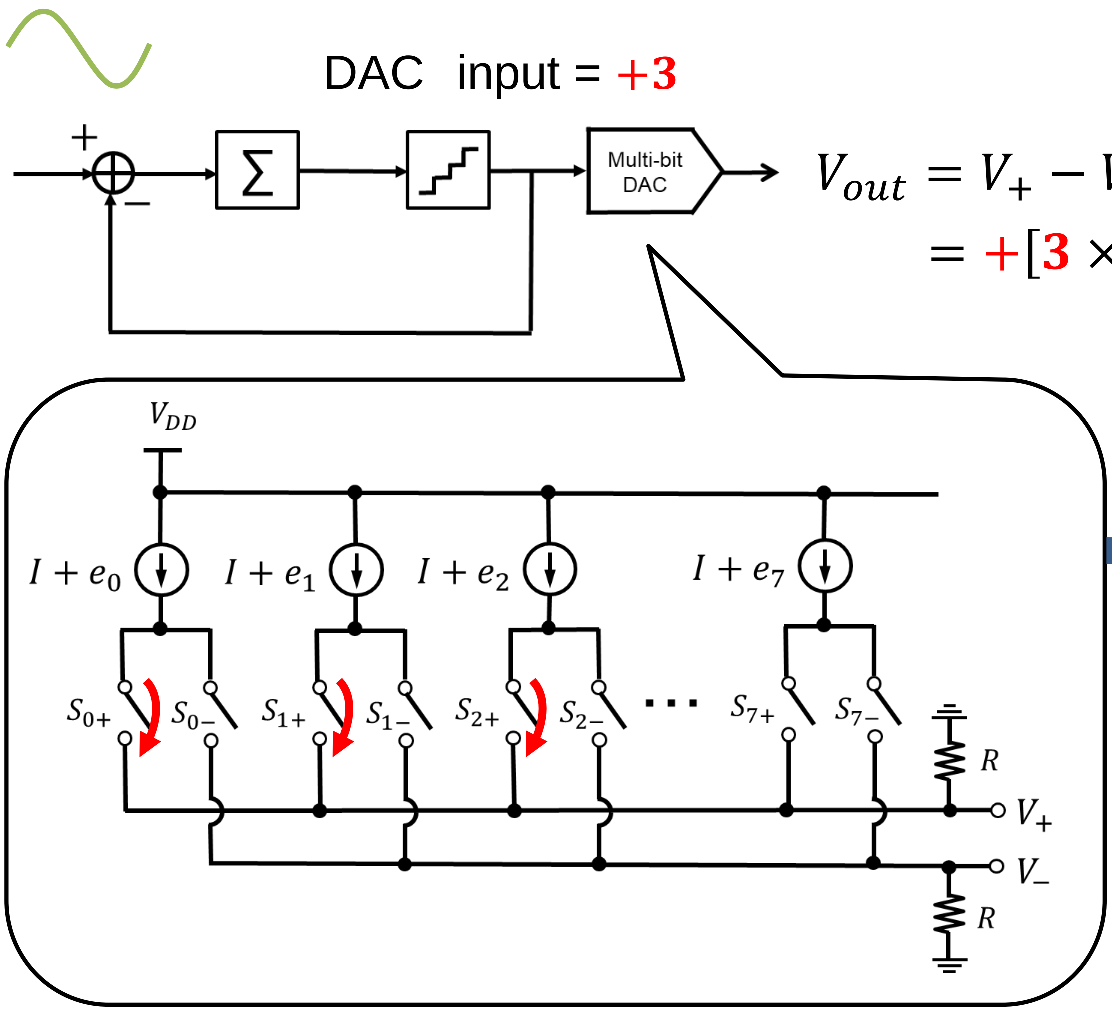
0 **+1** **-1** **0** **+1**

Binary unit cell → 1-bit

Ternary unit cell → 1.5-bit

Ternary → Higher resolution for given current sources
Smaller number of current sources for given resolution

Multi-bit DAC Operation of Ternary (1/3)



$$V_{out} = V_+ - V_-$$
$$= +[3 \times I + e_0 + e_1 + e_2]\,R$$

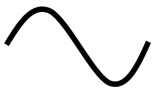
Positive voltage

Digital	V_{out}
+3	$+[3I + e_0 + e_1 + e_2]\,R$
0	
-2	

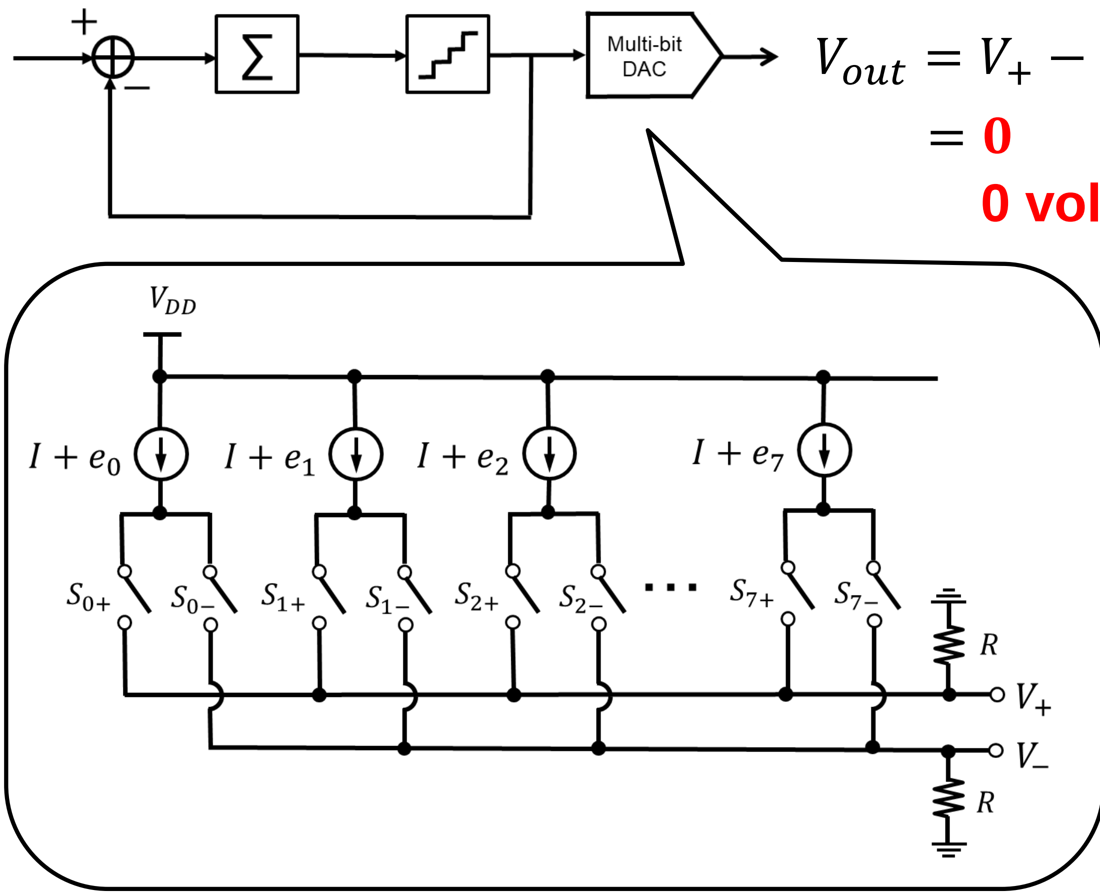
current $I_k = I + e_i$

e_i : Variation in current cell

Multi-bit DAC Operation of Ternary (2/3)



DAC input = **0**



$$V_{out} = V_+ - V_-$$
$$= \mathbf{0}$$

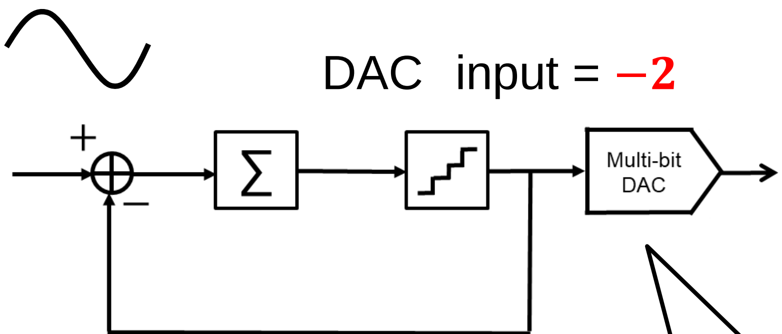
0 voltage

Digital	V_{out}
+3	$+[3I + e_0 + e_1 + e_2]\,R$
0	**0**
-2	

current $I_k = I + e_i$

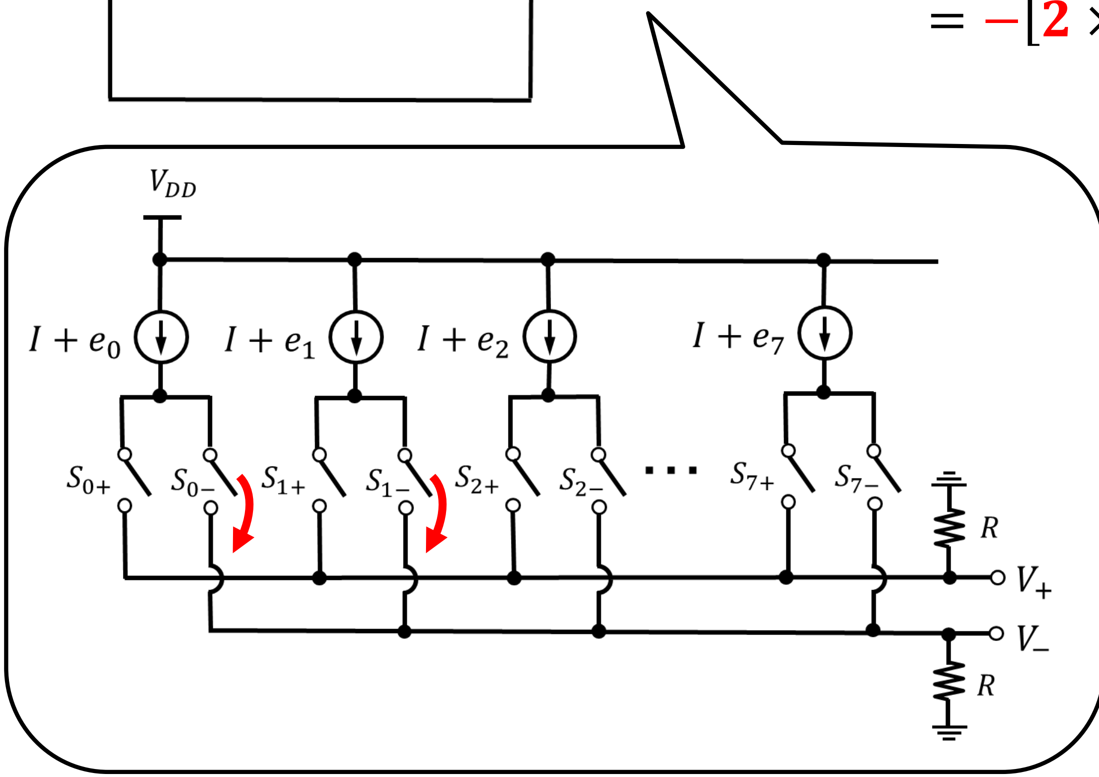
e_i : Variation in current cell

Multi-bit DAC Operation of Ternary (3/3)



$$V_{out} = V_+ - V_-$$
$$= -[2 \times I + e_0 + e_1]\, R$$

Negative voltage

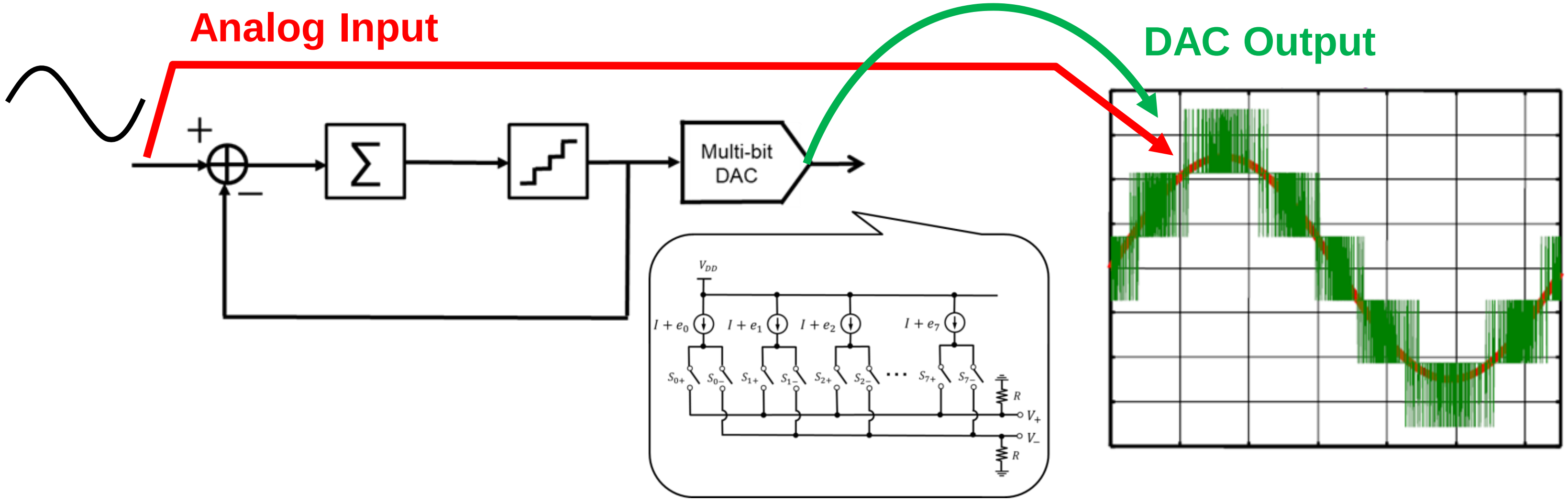


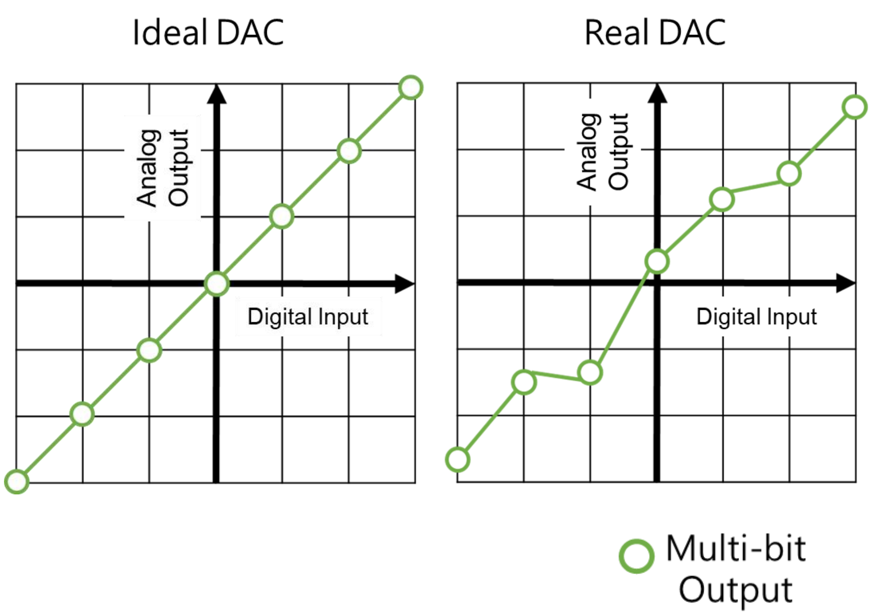
Digital	V_{out}
+3	$+[3I + e_0 + e_1 + e_2]\, R$
0	0
-2	$-[2 \times I + e_0 + e_1]\, R$

current $I_k = I + e_i$

e_i : Variation in current cell

Nonlinearity Problem of Multi-bit ΔΣ DAC





Merit

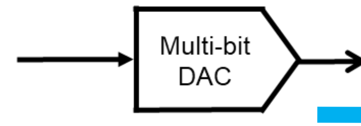
- Quantization error ⇒ Decrease
- Performance of following analog filter ⇒ Ease

Nonlinearity Problem

- Characteristics mismatches among multiple unit cell ⇒ **Nonlinearity problem**

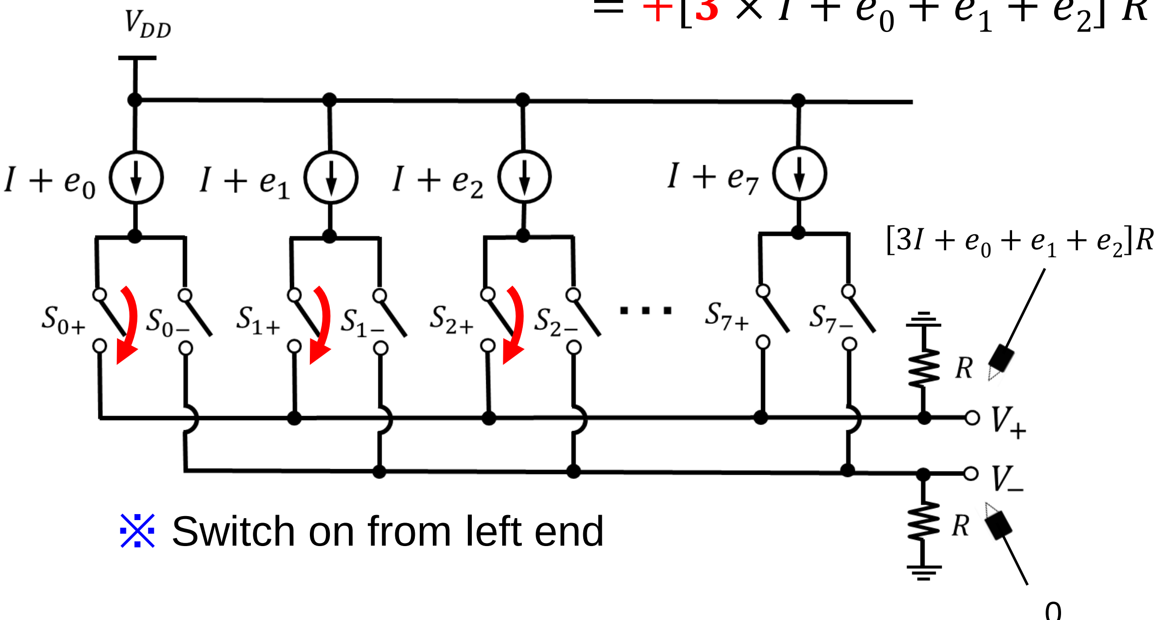
Multi-bit DAC of Ternary (1/2)

◆ Segment type DAC of ternary

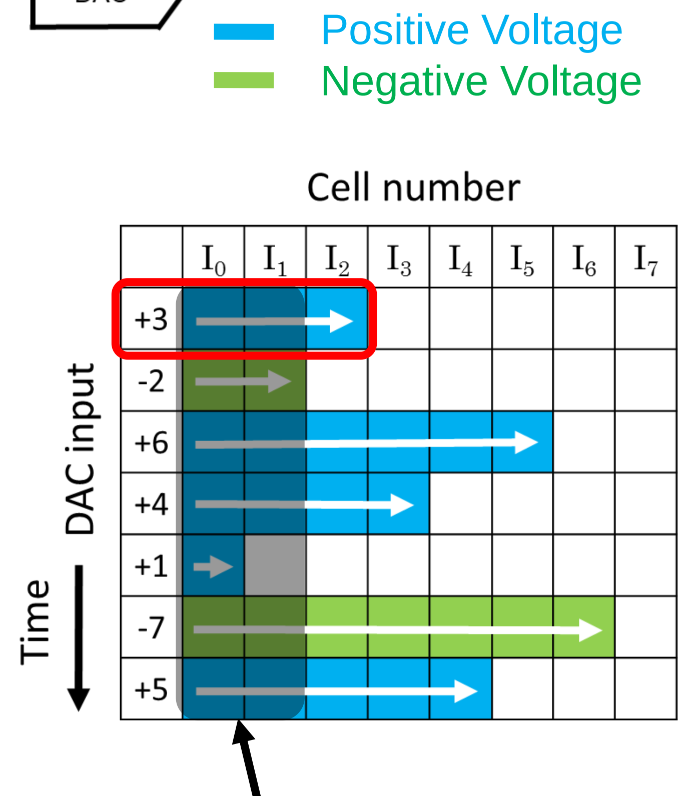


DAC input = +3 $\quad V_{out} = V_+ - V_-$

$$= +[3 \times I + e_0 + e_1 + e_2]\, R$$



※ Switch on from left end



Current source in DAC

✔Ideal ⇒ All equal

✔Real ⇒ Process variation on manufacturing

Unit cell mismatch
⇒ Accumulation
e_i

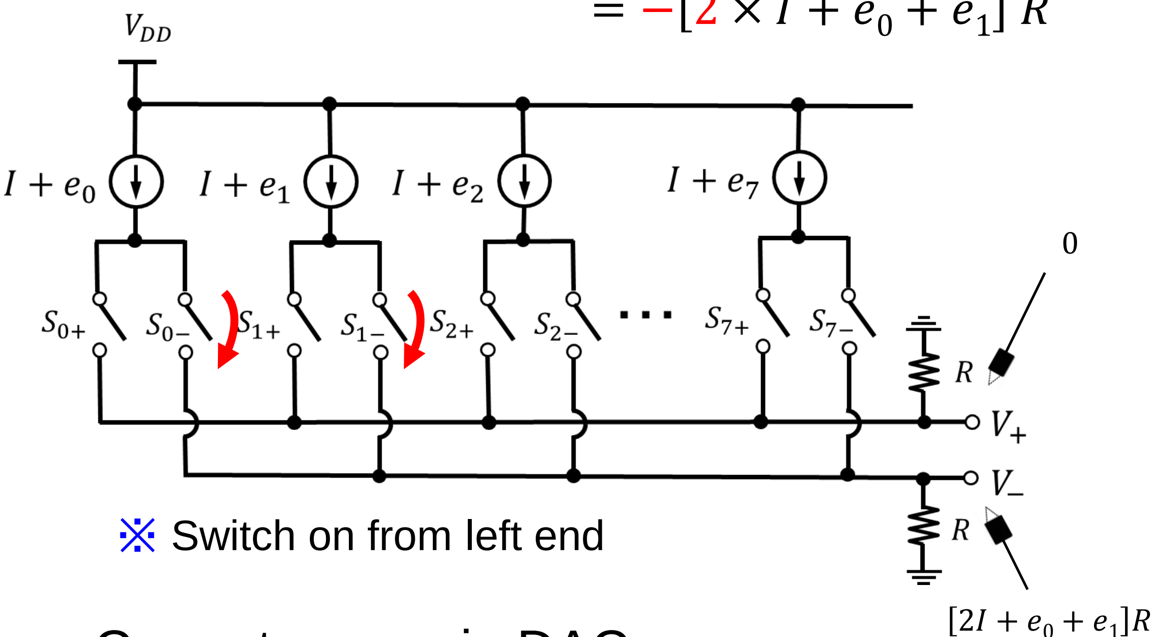
(current $\quad I_k = I + e_i$)

Multi-bit DAC of Ternary (2/2)

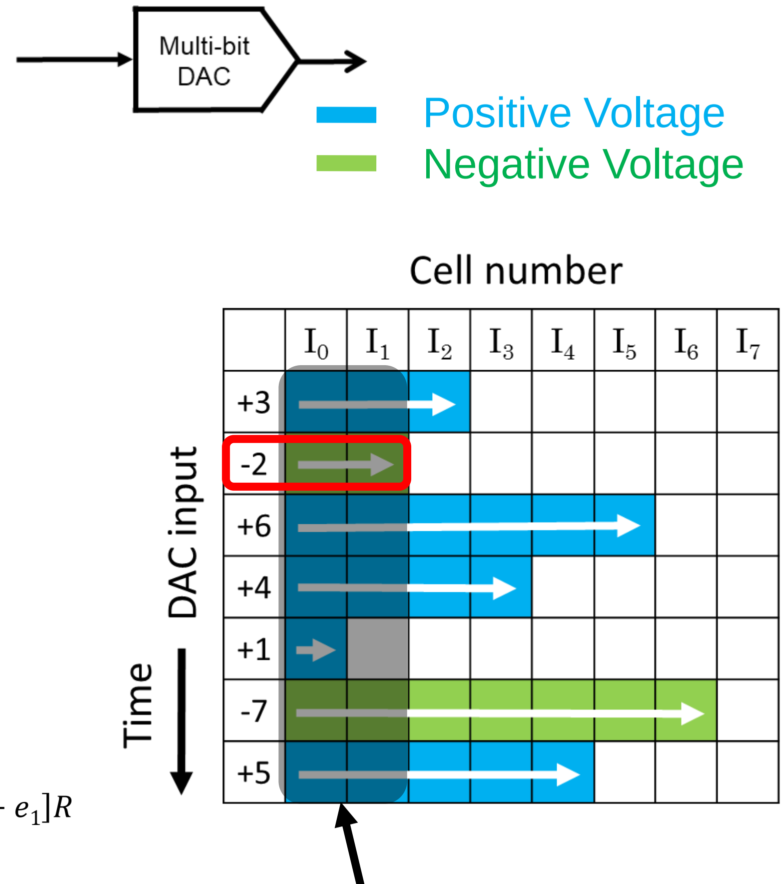
◆ Segment type DAC of ternary

DAC input = -2 $V_{out} = V_+ - V_-$

$= -[2 \times I + e_0 + e_1]\,R$



※ Switch on from left end



Current source in DAC

✔Ideal ⇒ All equal

✔Real ⇒ Process variation on manufacturing

(current $I_k = I + e_i$)

Unit cell mismatch
⇒ Accumulation
e_i

Outline

- Research Background
- ΔΣDA Converter
  - DWA*Algorithm　(* Data-Weighted Averaging)
- Simulation Verification
  - Binary, Ternary DWA Overview
  - ΔΣ DA Converter : HP type
  - ΔΣ DA Converter : BP type
- Conclusion

Unit Cell Current Average I

Unit cell current average : $I = \frac{1}{8}(I_0 + I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7)$

$I_0 = I + e_0$

$I_1 = I + e_1$

$I_2 = I + e_2$

$I_3 = I + e_3$

$I_4 = I + e_4$

$I_5 = I + e_5$

$I_6 = I + e_6$

$I_7 = I + e_7$

mismatch : e_k

→ total sum

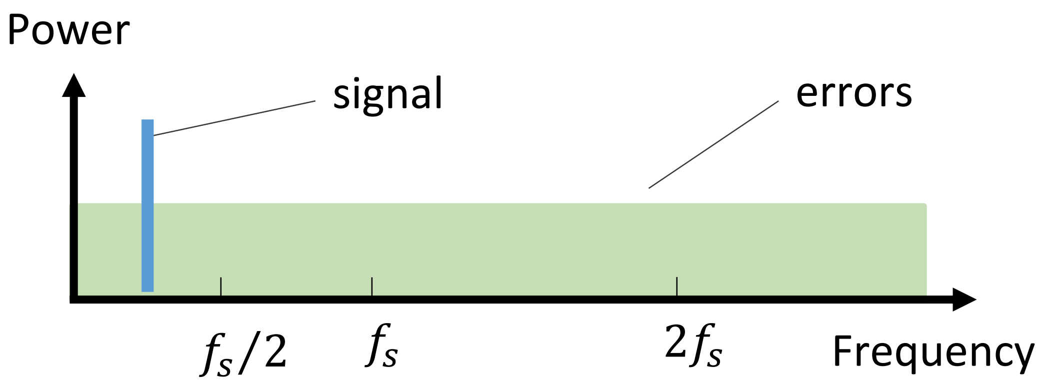
$$(I_0 + I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7) = 4I + (e_0 + e_1 + e_2 + e_3 + e_4 + e_5 + e_6 + e_7)$$

$$I = \frac{1}{4}(I_0 + I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7) - \frac{1}{4}(e_0 + e_1 + e_2 + e_3 + e_4 + e_5 + e_6 + e_7)$$

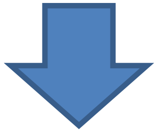
$(e_0 + e_1 + e_2 + e_3 + e_4 + e_5 + e_6 + e_7)$ → 0

DWA Techniques

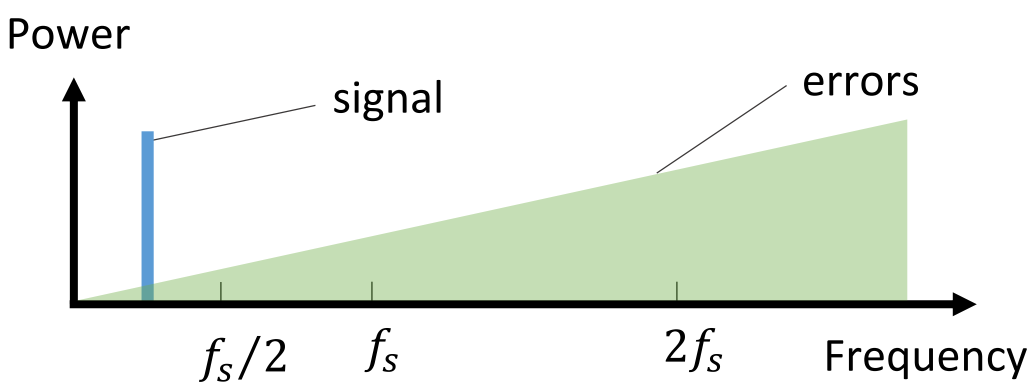
w/o DWA



- Errors are flat in spectrum



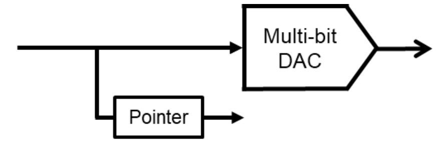
w/ DWA LP type



- Spectrum of errors are low-pass shaped.

Multi-bit DAC of Ternary and DWA I (1/2)

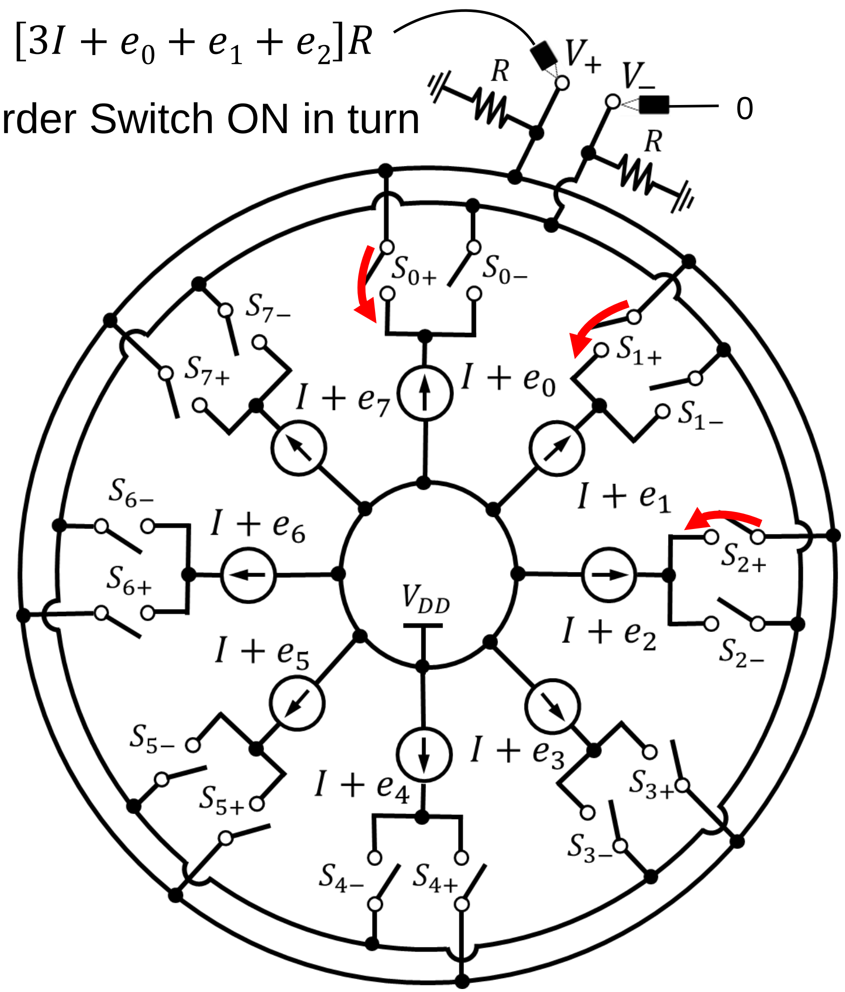
◆ DWA* type DAC of ternary (*Data-Weighted Average)



DAC input = +3

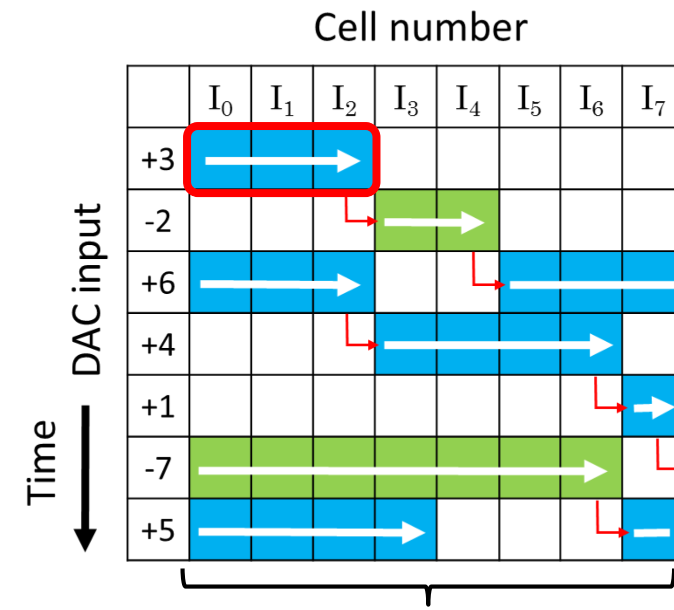
$$V_{out} = V_+ - V_-$$
$$= +[3 \times I + e_0 + e_1 + e_2]\, R$$

※ In order Switch ON in turn



Positive Voltage

negative Voltage



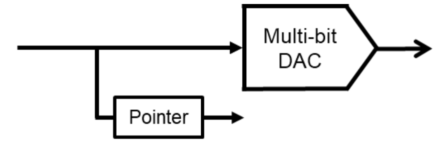
Distribute unit cell mismatch

⇒ Averaging

Improve linearity

Multi-bit DAC of Ternary and DWA I (2/2)

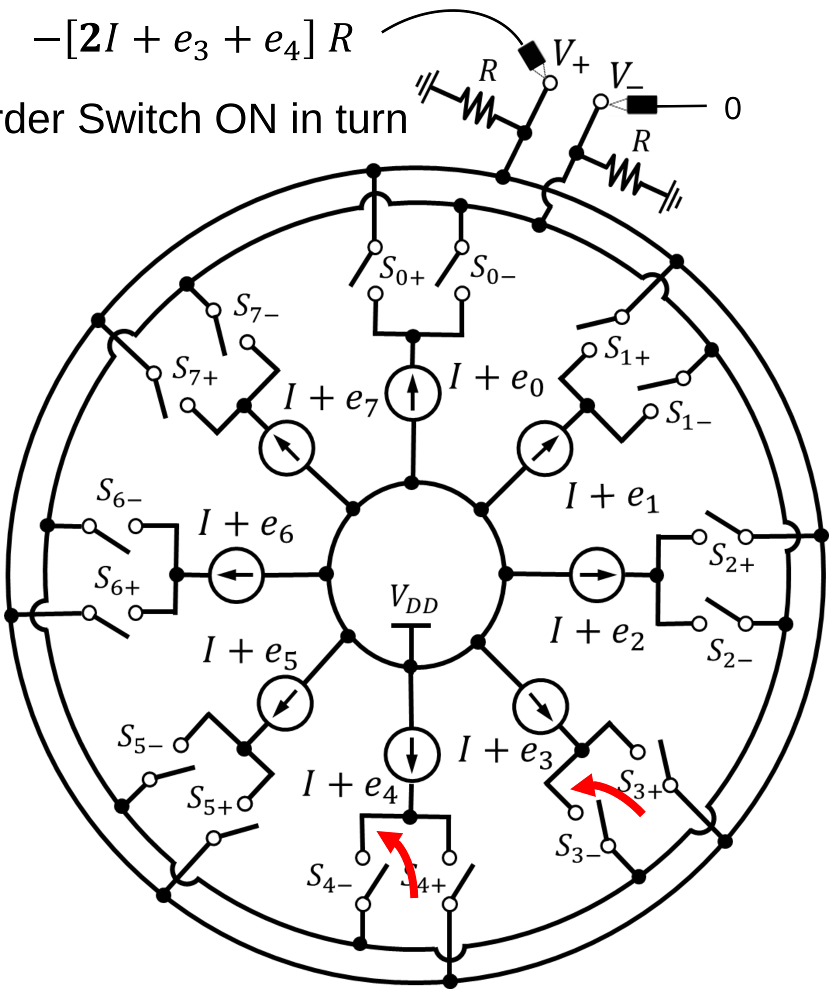
◆ DWA* type DAC of ternary (*Data-Weighted Average)

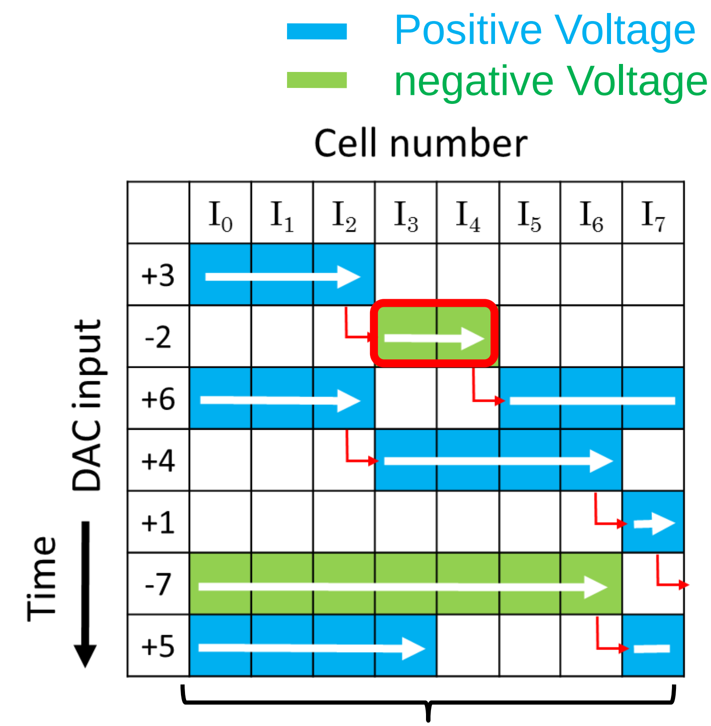


DAC input = **−2**

$$V_{out} = V_+ - V_-$$
$$= -[\mathbf{2} \times I + e_3 + e_4]\,R$$

※ In order Switch ON in turn





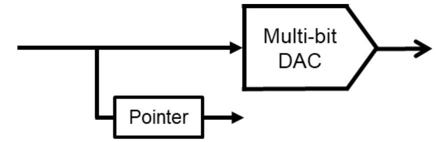
Distribute unit cell mismatch

⇒ Averaging

Improve linearity

Multi-bit DAC of Ternary and DWA II (1/3)

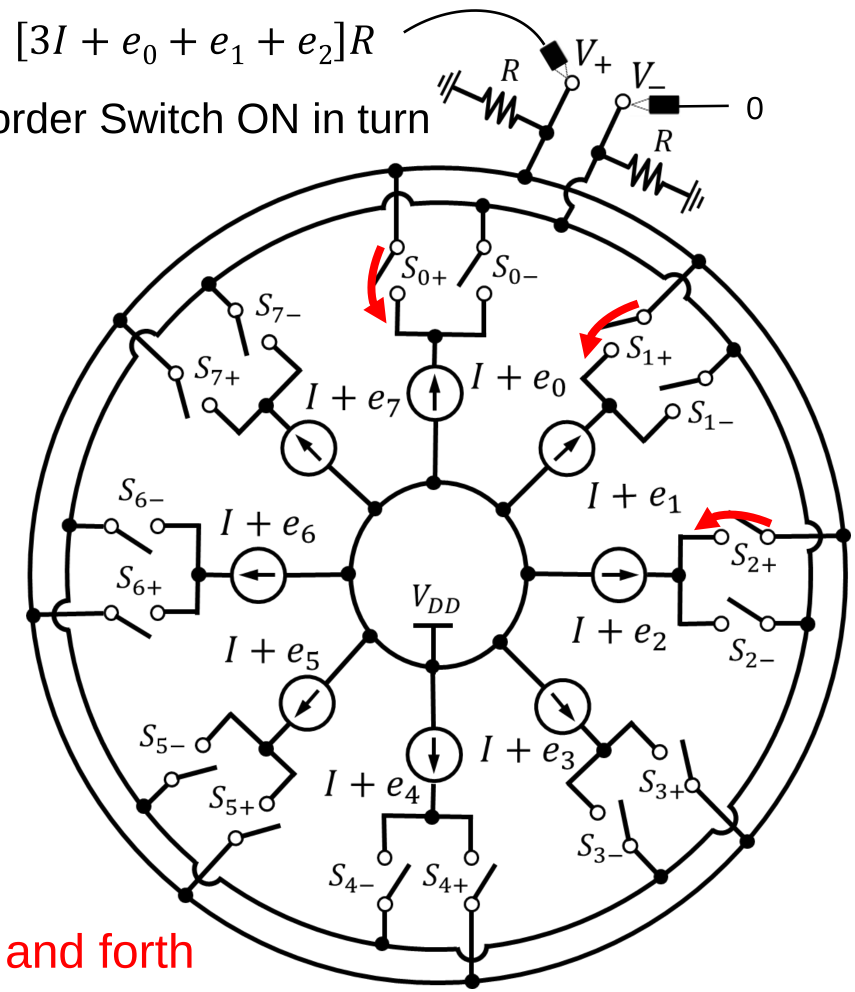
◆ DWA* type DAC of ternary (*Data-Weighted Average)



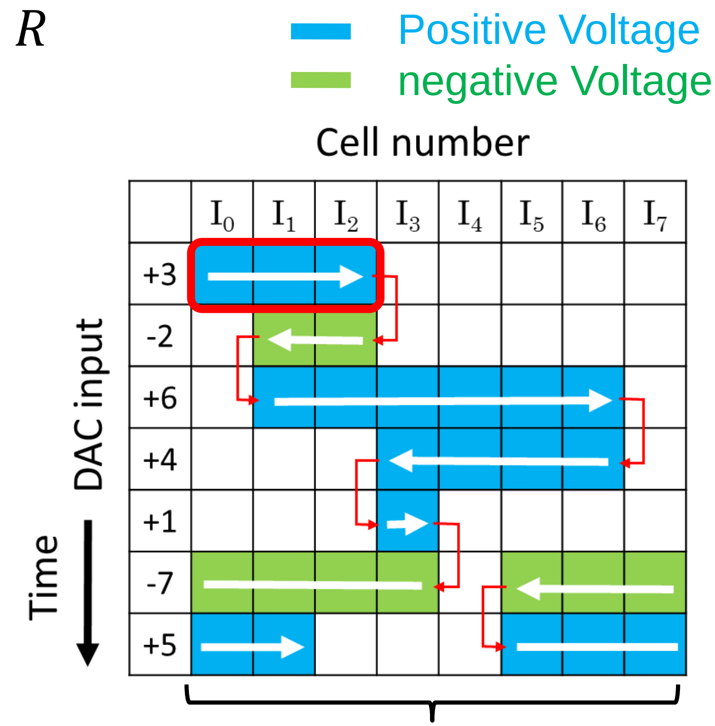
DAC input = +3

$$V_{out} = V_+ - V_-$$
$$= +[3 \times I + e_0 + e_1 + e_2]\,R$$

※ In order Switch ON in turn



Back and forth



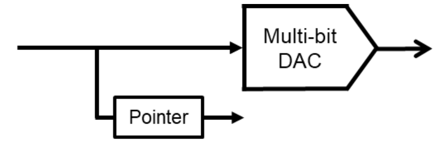
Distribute unit cell mismatch

⇒ Averaging

Improve linearity

Multi-bit DAC of Ternary and DWA II (2/3)

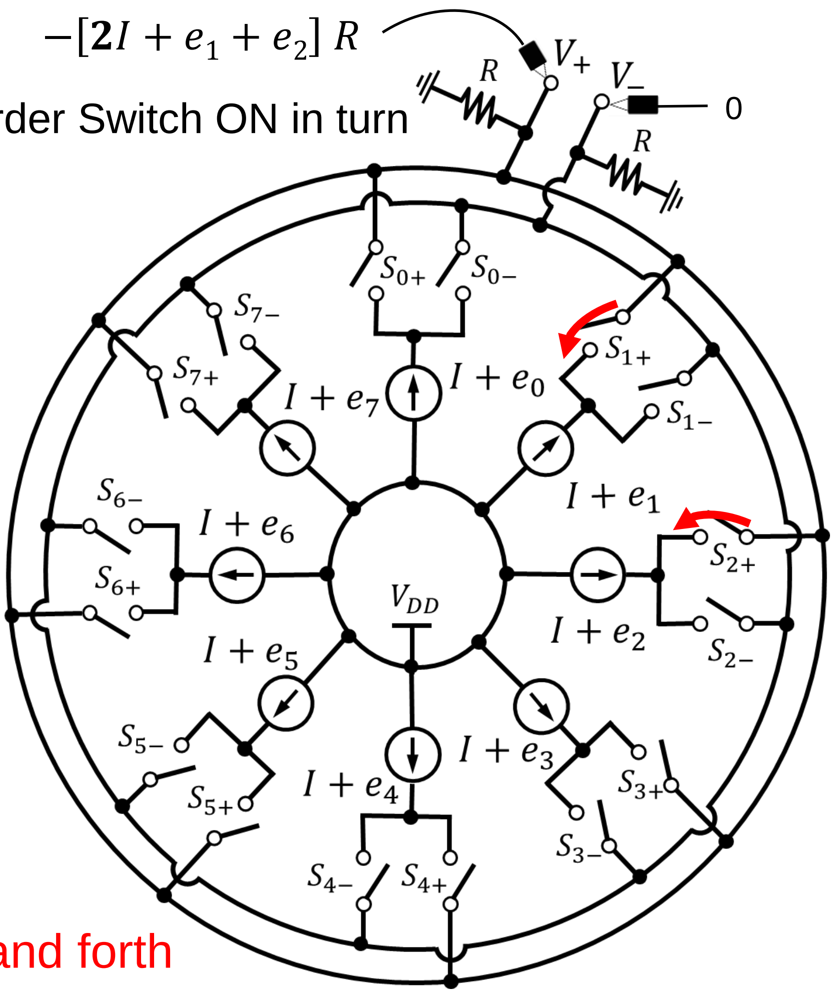
◆ DWA* type DAC of ternary (*Data-Weighted Average)



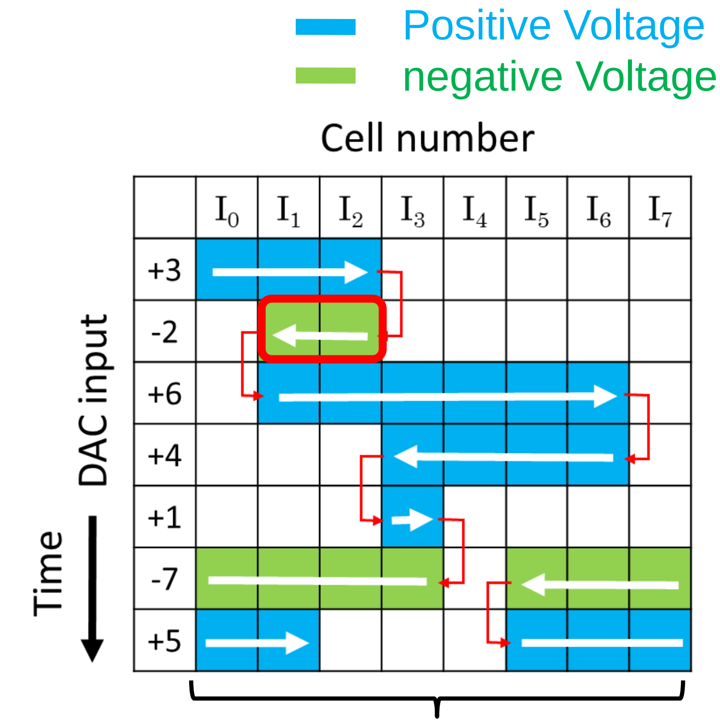
DAC input = -2

$$V_{out} = V_+ - V_- = -[2 \times I + e_1 + e_2]\,R$$

※ In order Switch ON in turn



Back and forth



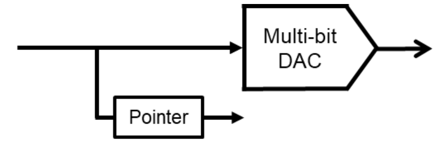
Distribute unit cell mismatch

⇒ Averaging

Improve linearity

Multi-bit DAC of Ternary and DWA II (3/3)

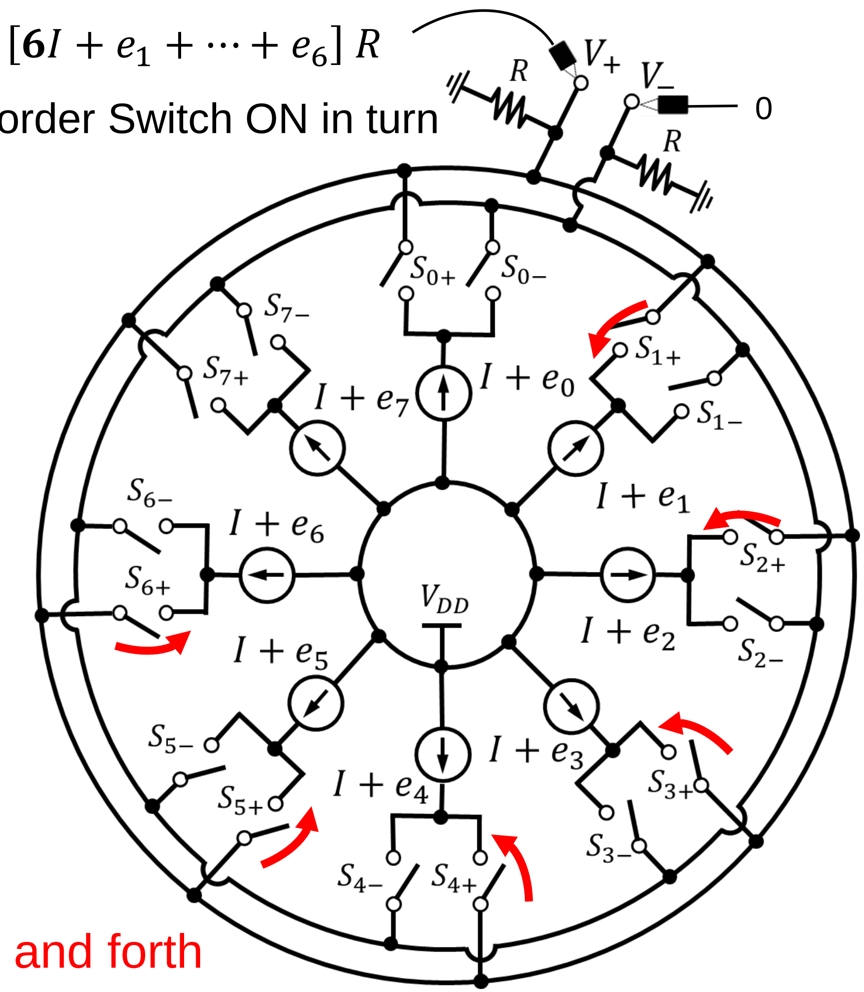
◆ DWA* type DAC of ternary (*Data-Weighted Average)



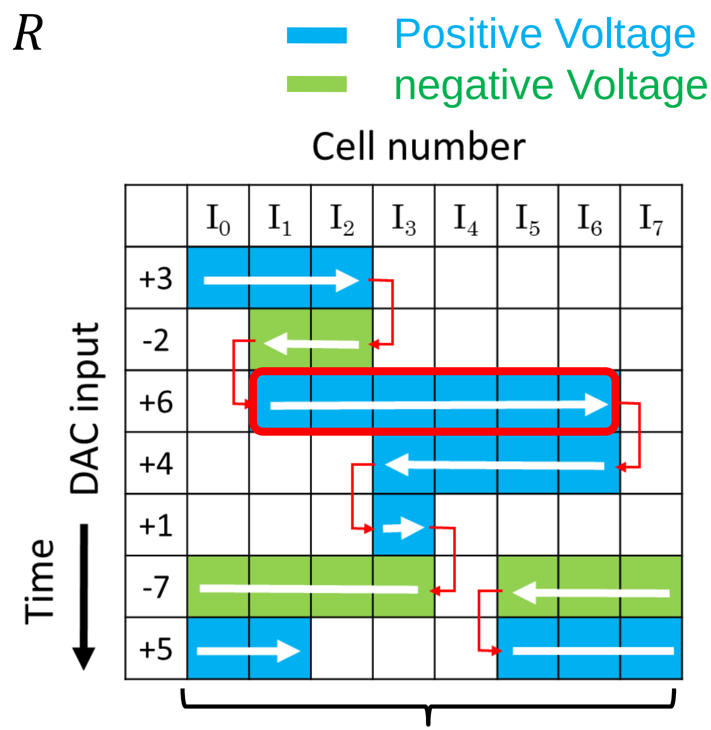
DAC input = +6

$$V_{out} = V_+ - V_-$$
$$= +[6 \times I + e_1 + \cdots + e_6]\, R$$

※ In order Switch ON in turn



Back and forth



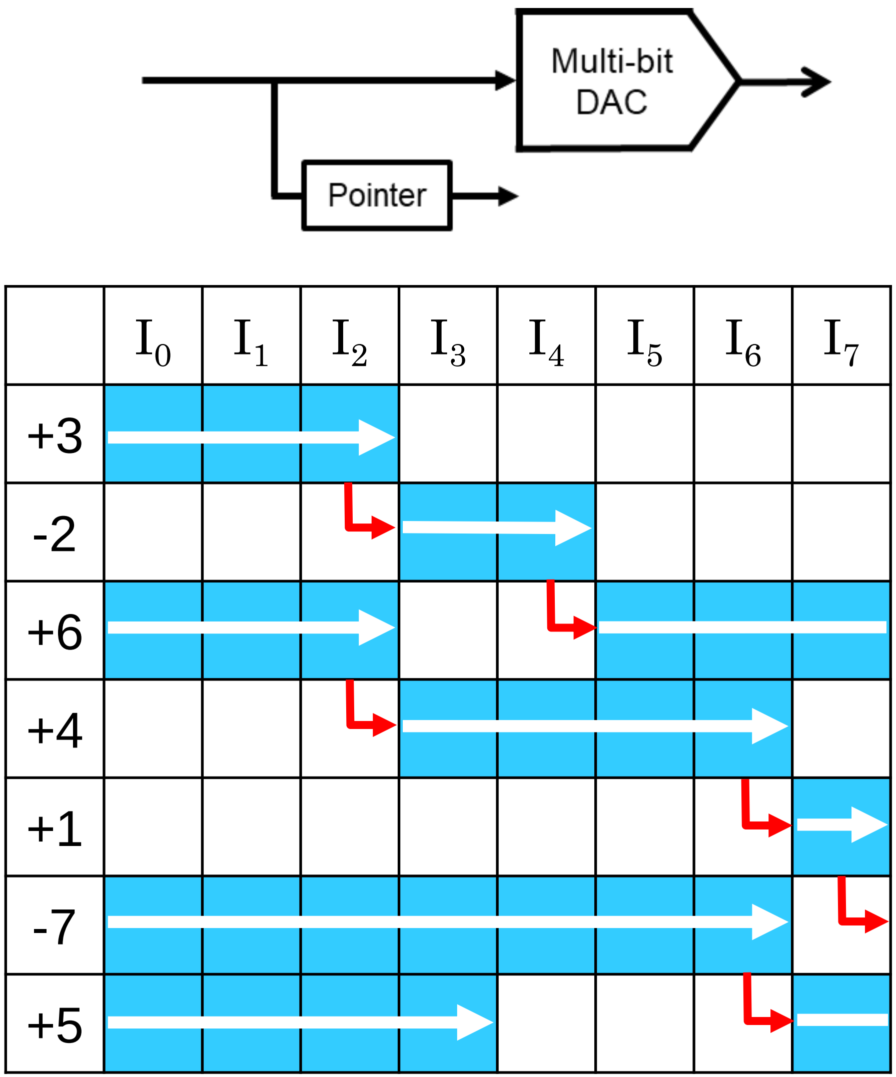
Distribute unit cell mismatch

⇒ Averaging

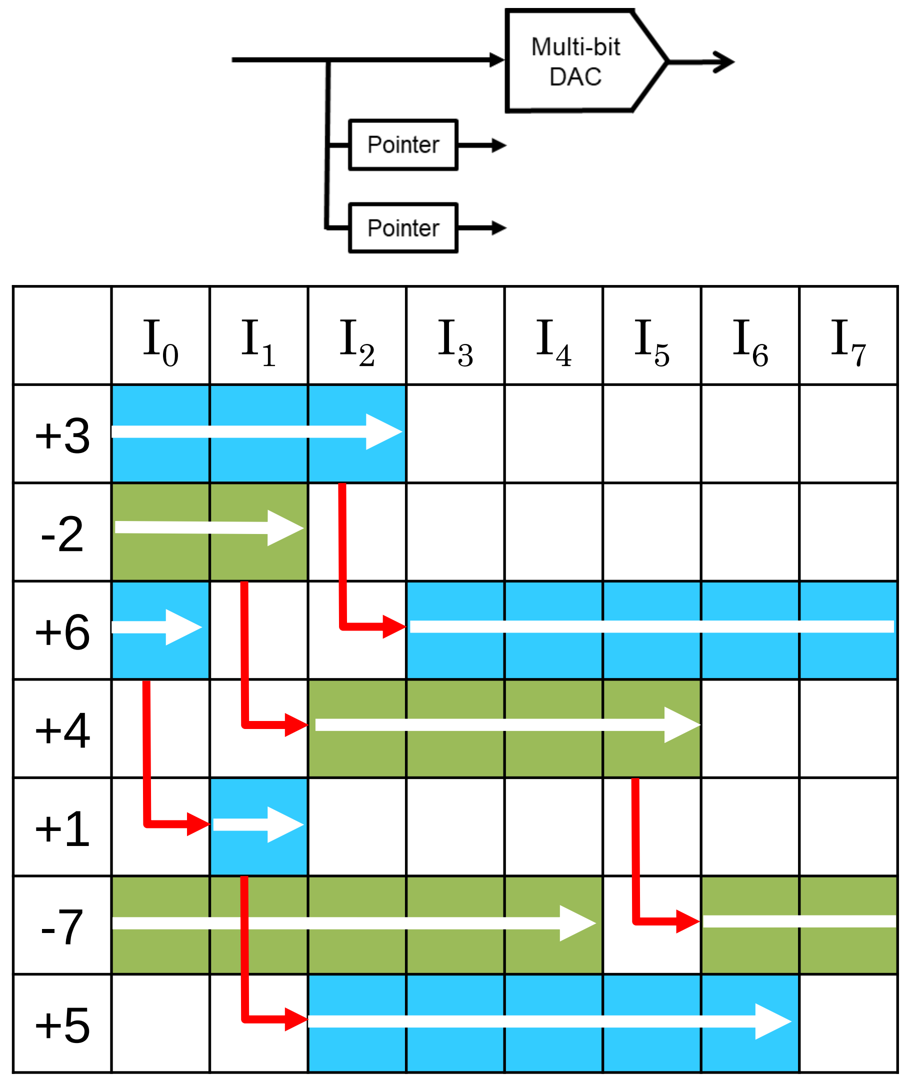
Improve linearity

DWA type I (Pointer)

1 Pointer

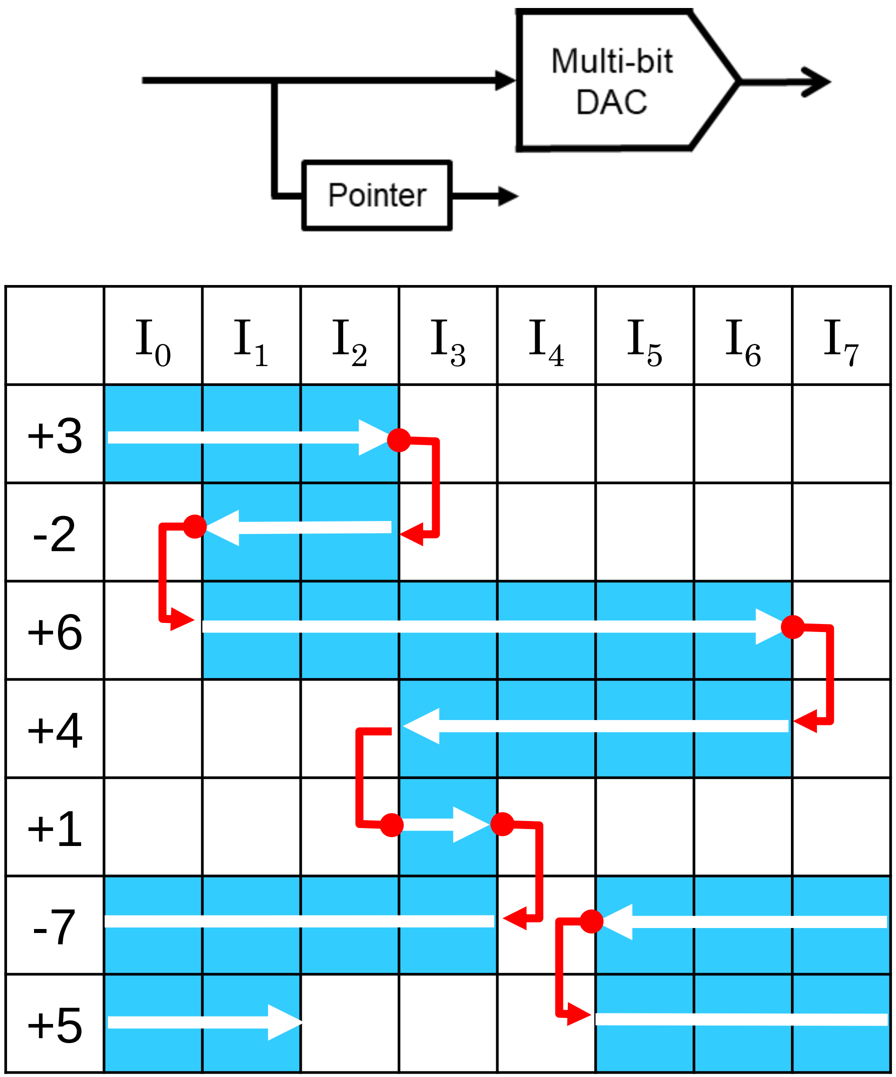


2 Pointers

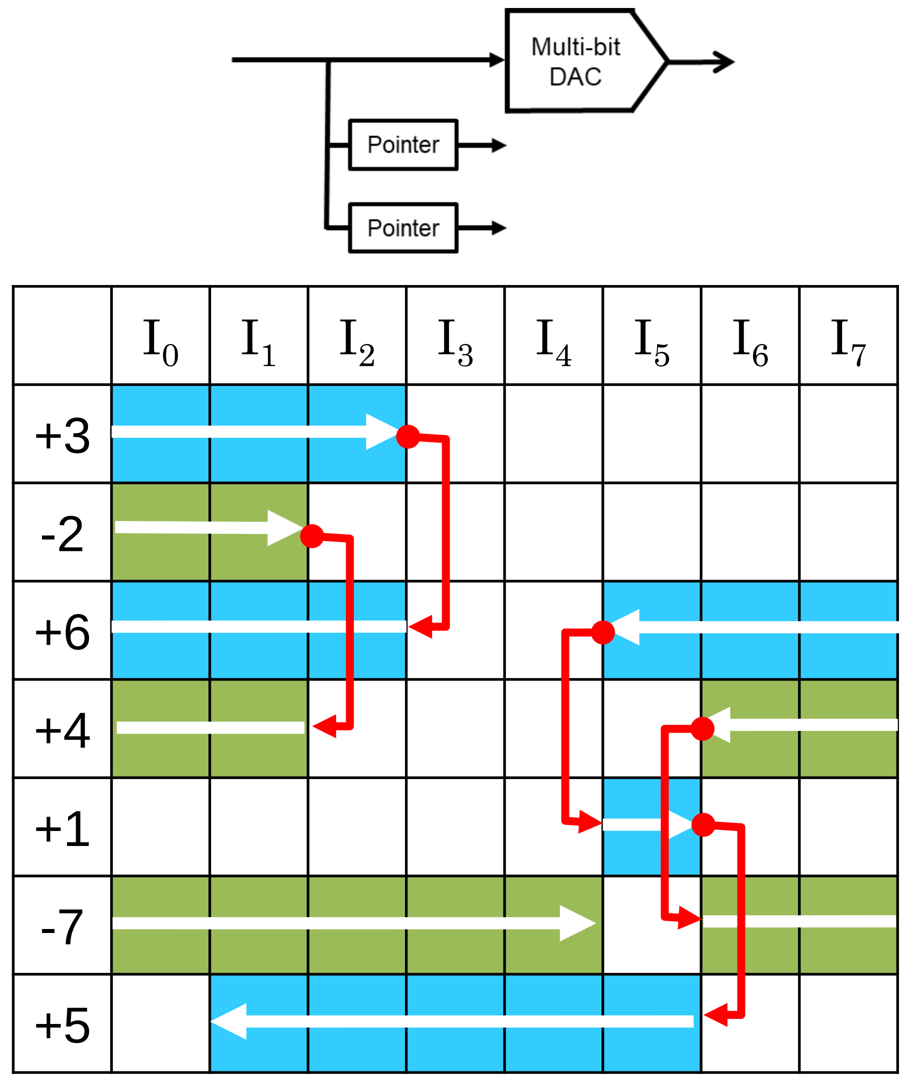


DWA type II (Pointer)

1 Pointer



2 Pointers



Outline

- Research Background
- ΔΣDA Converter
  - DWA*Algorithm (* Data-Weighted Averaging)
- Simulation verification
  - Binary, Ternary DWA Overview
  - ΔΣ DA Converter : HP type
  - ΔΣ DA Converter : BP type
- Conclusion

Binary, Ternary DWA Overview

Signal Band	Value	Number (N) of Signal Bands	DWA type
LP	Binary	1	I
	Ternary	1	I
HP	Binary	1	II
	Ternary	1	I
BP	Binary	2	II
	Binary	4	II
	Ternary	2	I
	Ternary	4	I

New Findings

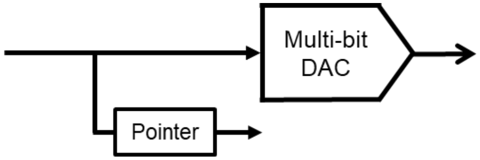
Outline

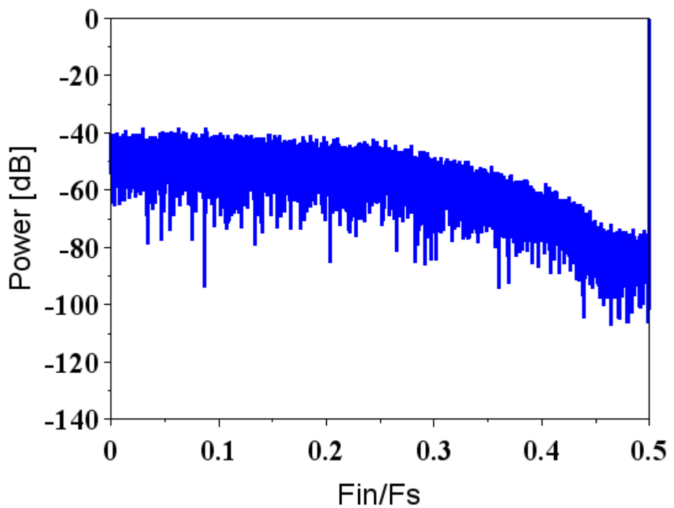
- Research Background
- ΔΣDA Converter
  - DWA*Algorithm (* Data-Weighted Averaging)
- Simulation verification
  - Binary, Ternary DWA Overview
  - ΔΣ DA Converter : HP type
  - ΔΣ DA Converter : BP type
- Conclusion

High-Pass ΔΣDAC (Binary)

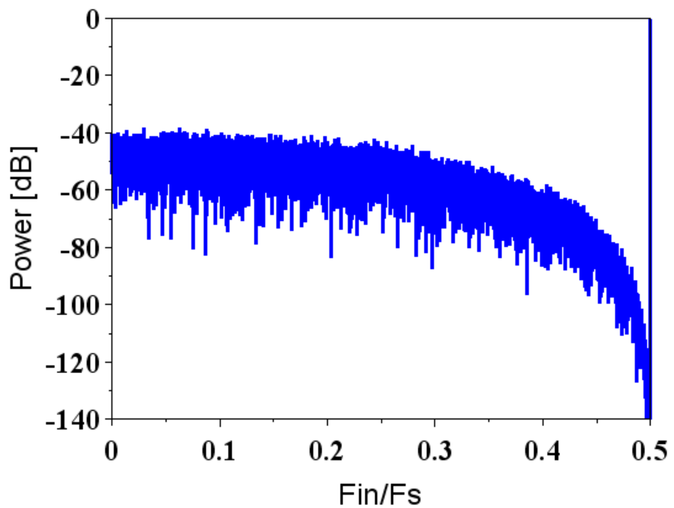
- High-pass (HP) ΔΣ DAC (N=1)

  Segmented DAC with **binary** unit cells





< DWA type I >

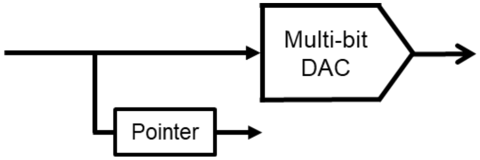


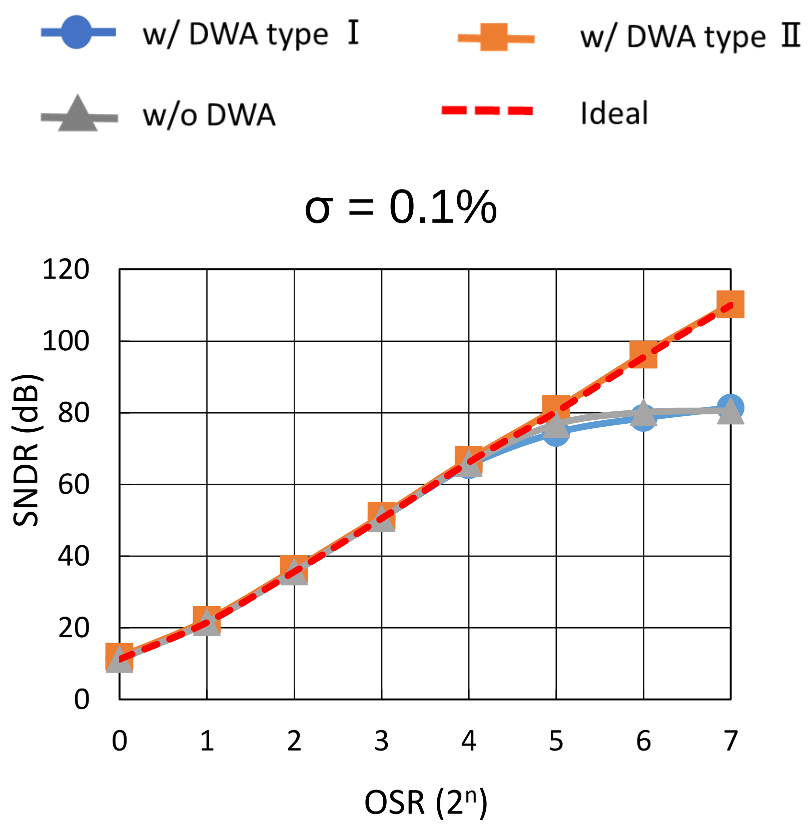
< DWA type II >

High-Pass ΔΣDAC (Binary)

- High-pass (HP) ΔΣ DAC (N=1)

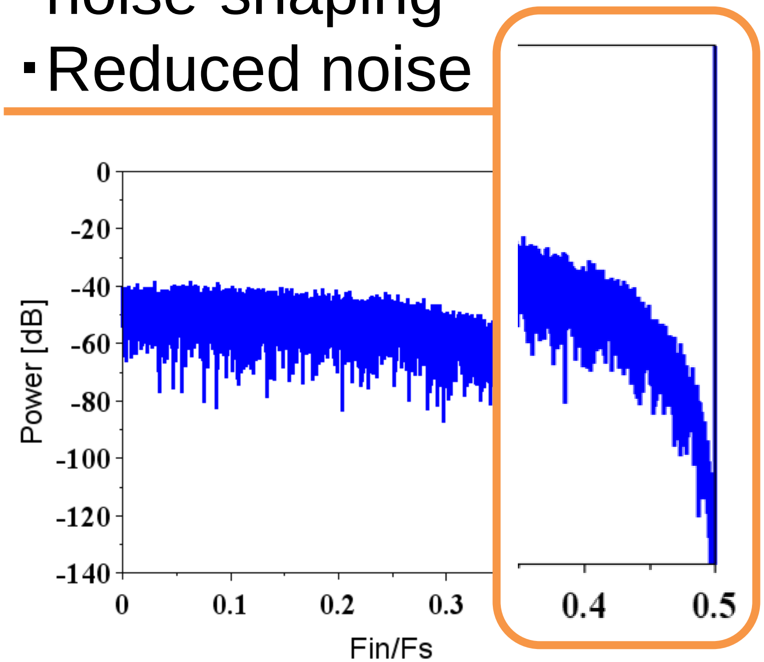
Segmented DAC with **binary** unit cells





Good Algorithm

- noise-shaping
- Reduced noise

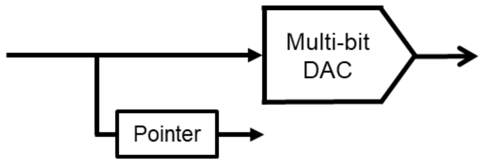


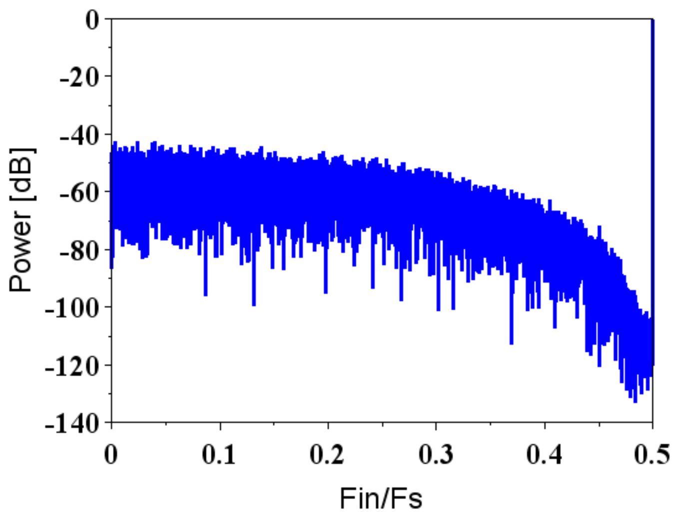
< DWA type II >

High-Pass ΔΣDAC (Ternary)

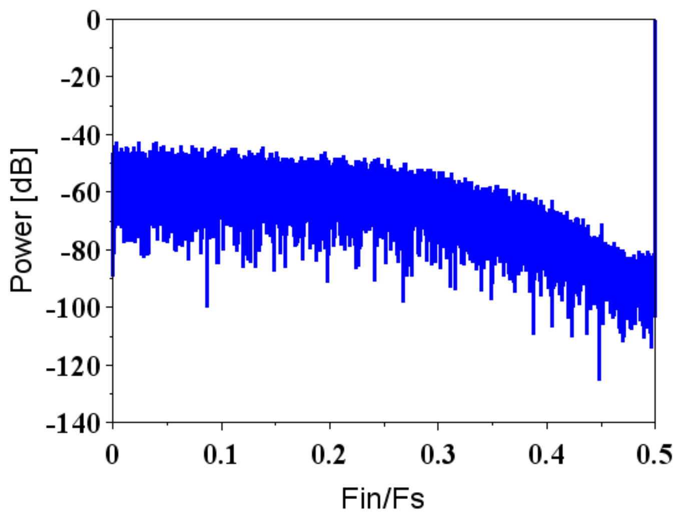
- High-pass (HP) ΔΣ DAC (N=1)

  Segmented DAC with **ternary** unit cells





< DWA type I >

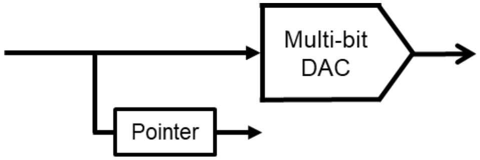


< DWA type II >

High-Pass ΔΣDAC (Ternary)

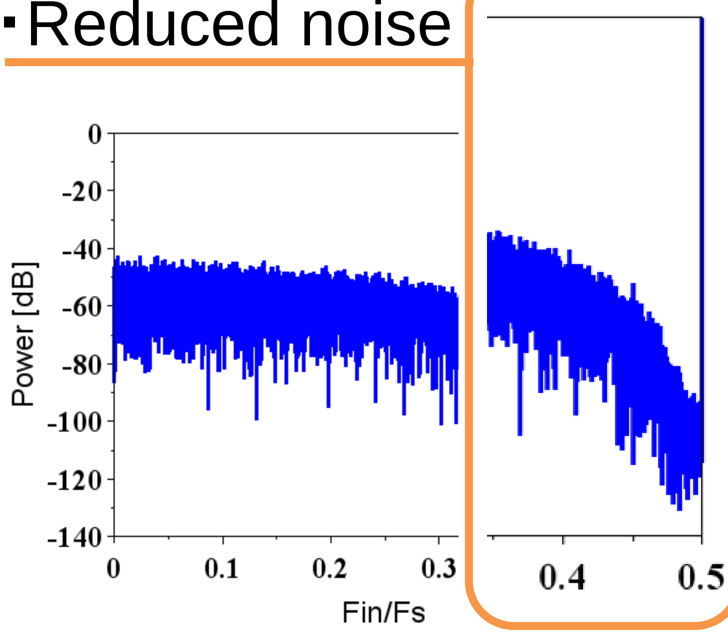
- High-pass (HP) ΔΣ DAC (N=1)

  Segmented DAC with ternary unit cells

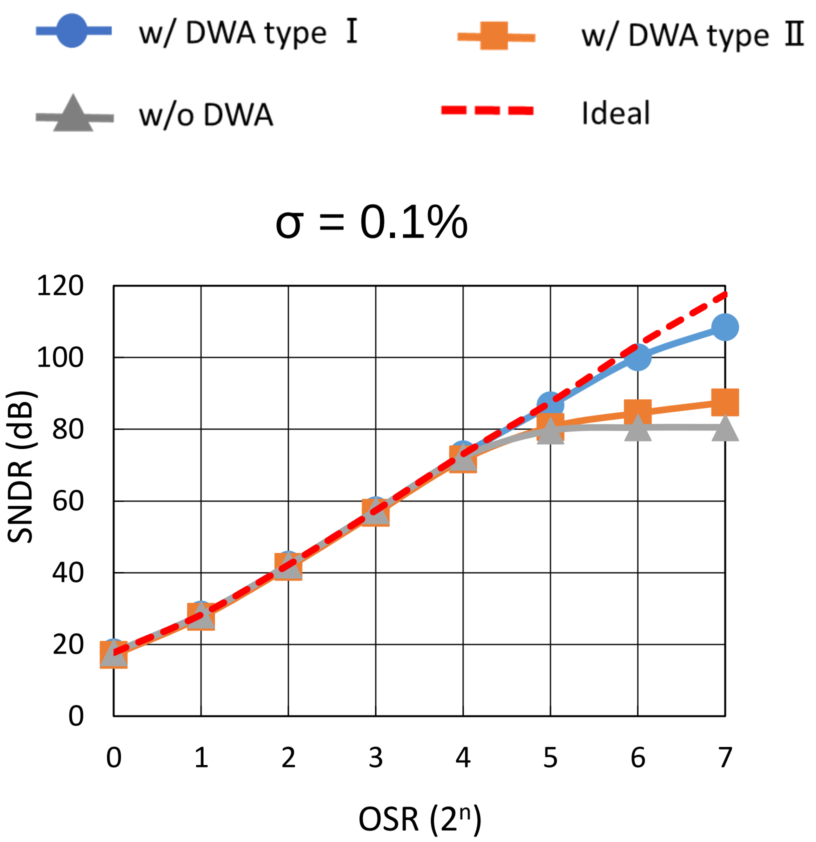


Good Algorithm

- noise-shaping
- Reduced noise



< DWA type I >



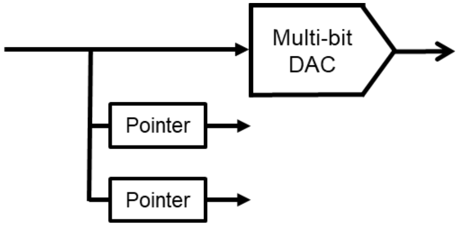
Outline

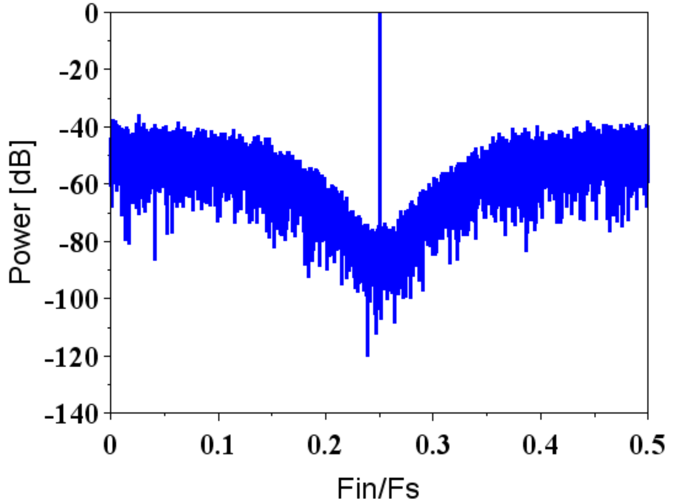
- Research Background
- ΔΣDA Converter
  - DWA*Algorithm (* Data-Weighted Averaging)
- Simulation verification
  - Binary, Ternary DWA Overview
  - ΔΣ DA Converter : HP type
  - ΔΣ DA Converter : BP type
- Conclusion

Band-Pass ΔΣDAC (Binary)

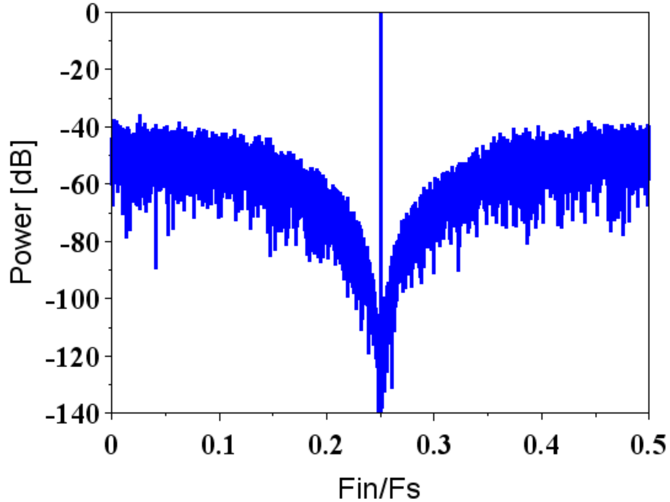
- Band-pass (BP) ΔΣ DAC (N=2)

  segmented DAC with **binary** unit cells





< DWA type I >

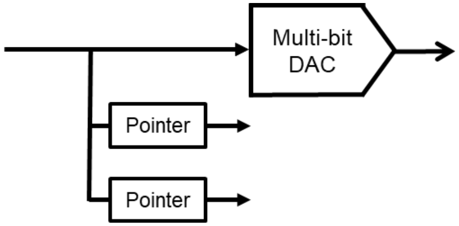


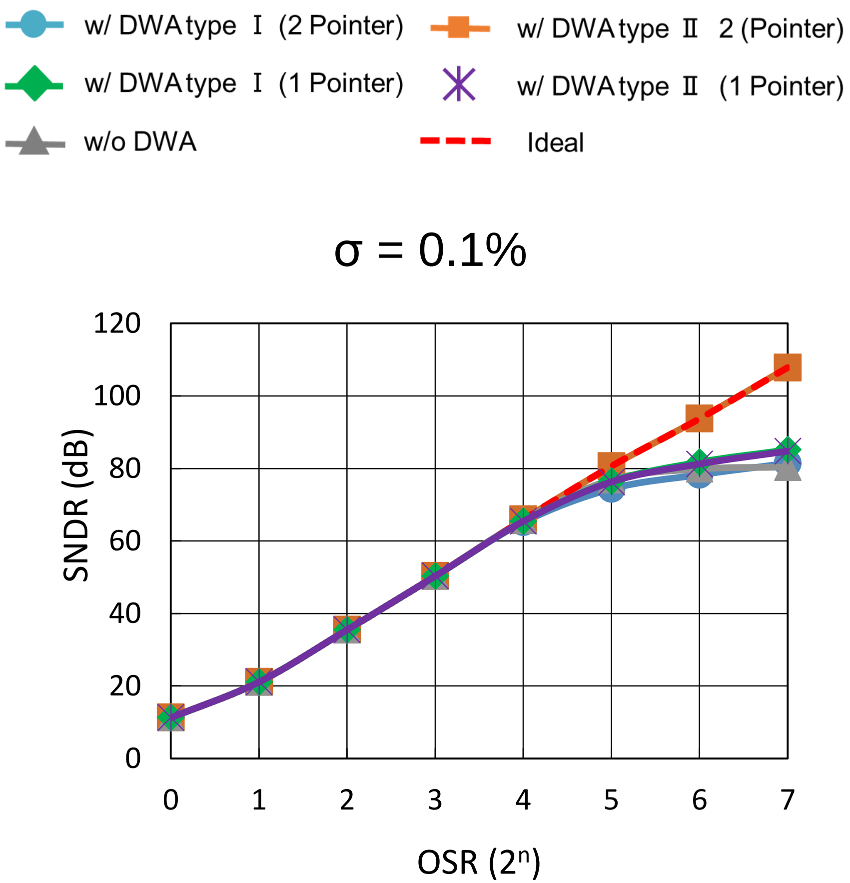
< DWA type II >

Band-Pass ΔΣDAC (Binary)

- Band-pass (BP) ΔΣ DAC (N=2)

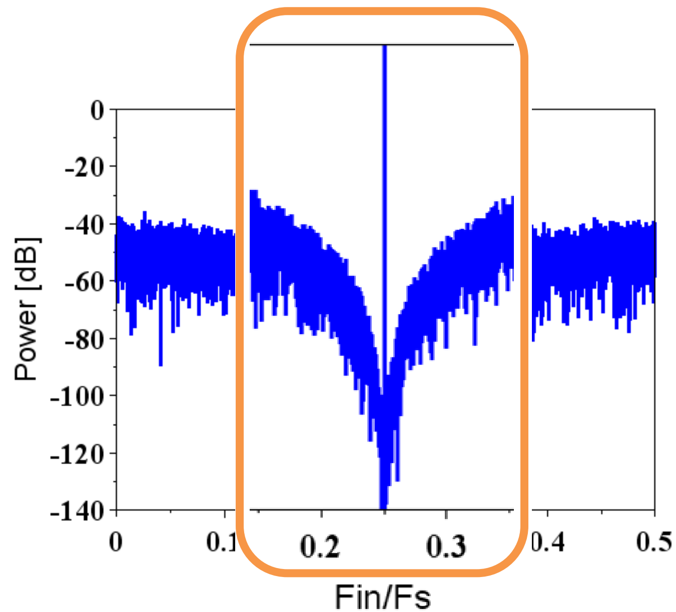
  Segmented DAC with **binary** unit cells





Good Algorithm

・noise-shaping

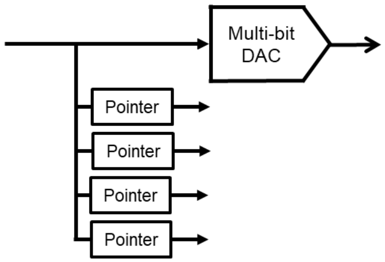


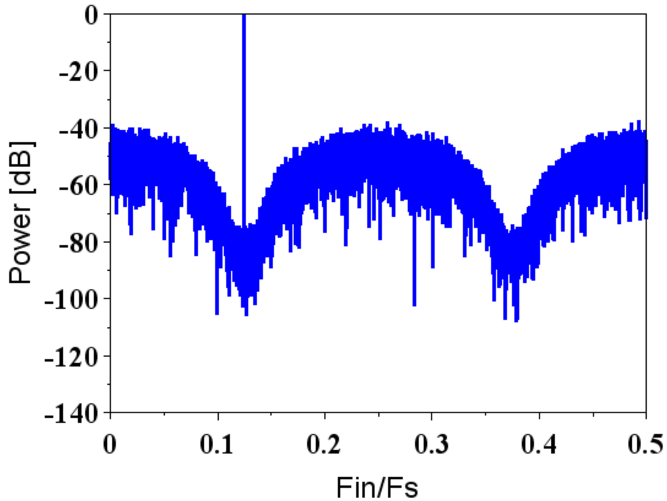
< DWA type II >

Band-Pass ΔΣDAC (Binary)

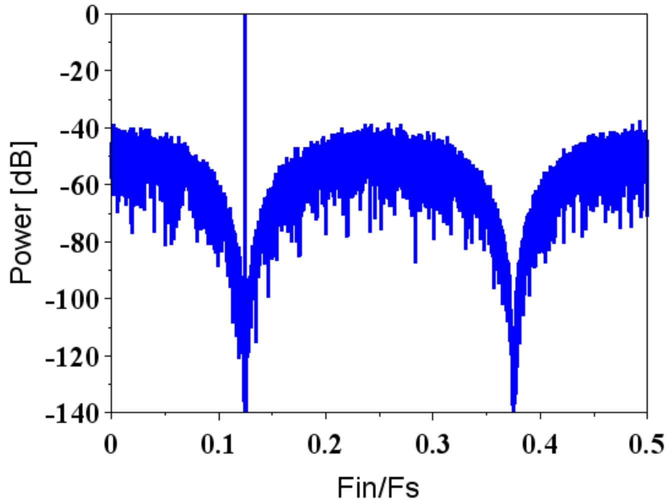
- Band-pass (BP) ΔΣ DAC (N=4)

  Segmented DAC with **binary** unit cells





< DWA type I >

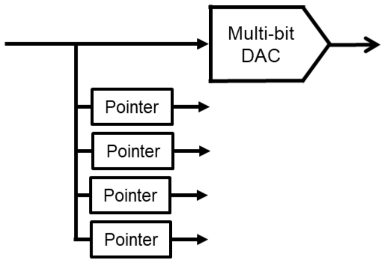


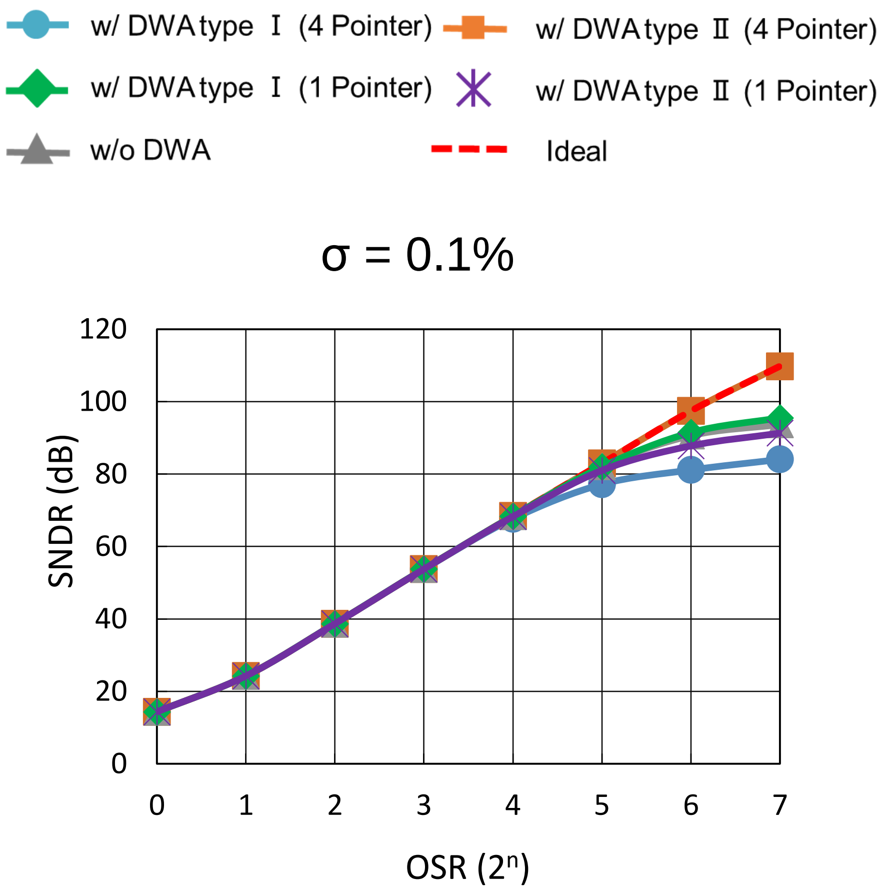
< DWA type II >

Band-Pass ΔΣDAC (Binary)

- Band-pass (BP) ΔΣ DAC (N=4)

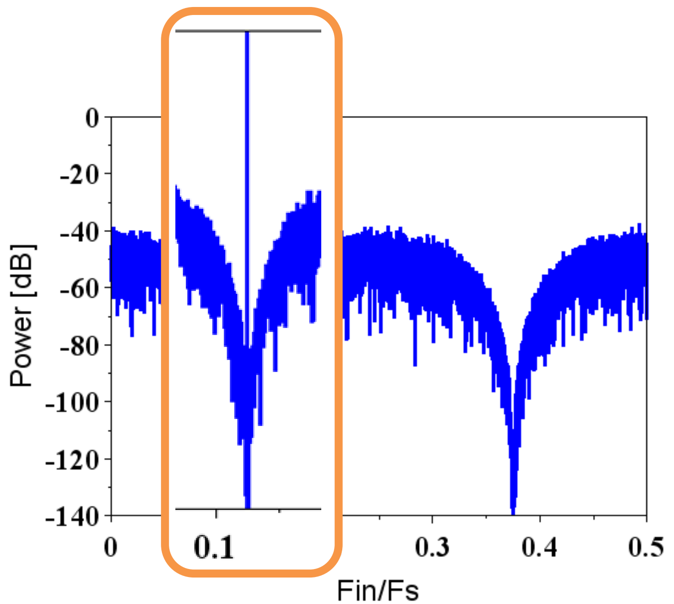
  Segmented DAC with **binary** unit cells





Good Algorithm

・noise-shaping

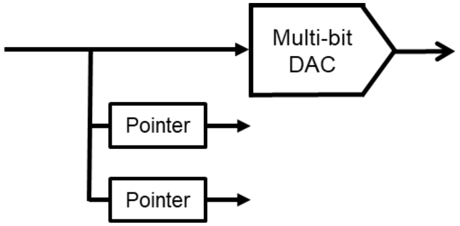


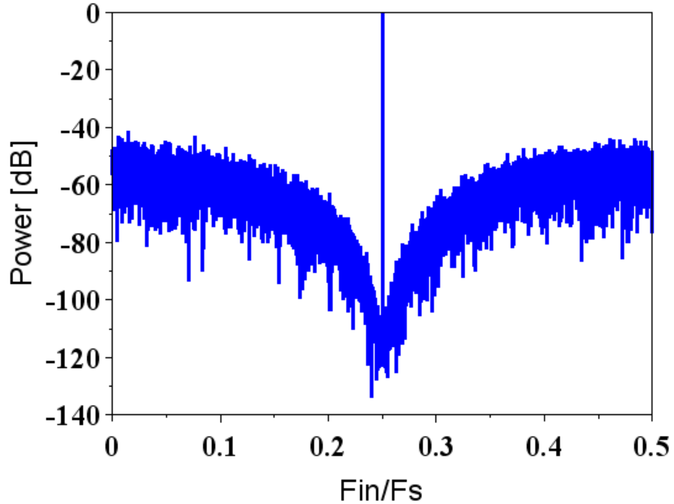
< DWA type II >

Band-Pass ΔΣDAC (Ternary)

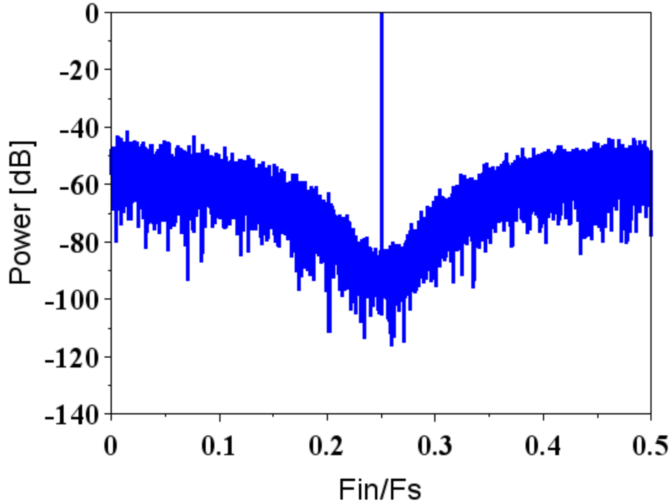
- Band-pass (BP) ΔΣ DAC (N=2)

  Segmented DAC with **ternary** unit cells





< DWA type I >

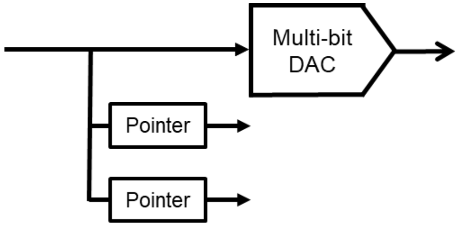


< DWA type II >

Band-Pass ΔΣDAC (Ternary)

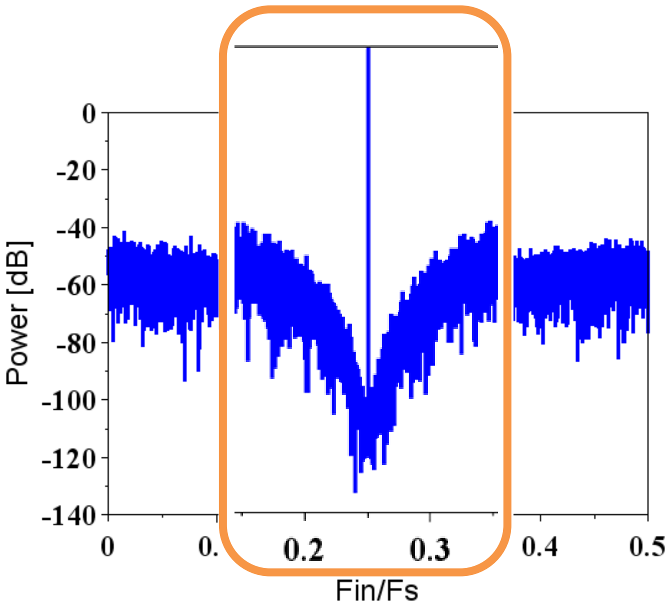
- Band-pass (BP) ΔΣ DAC (N=2)

  Segmented DAC with **ternary** unit cells

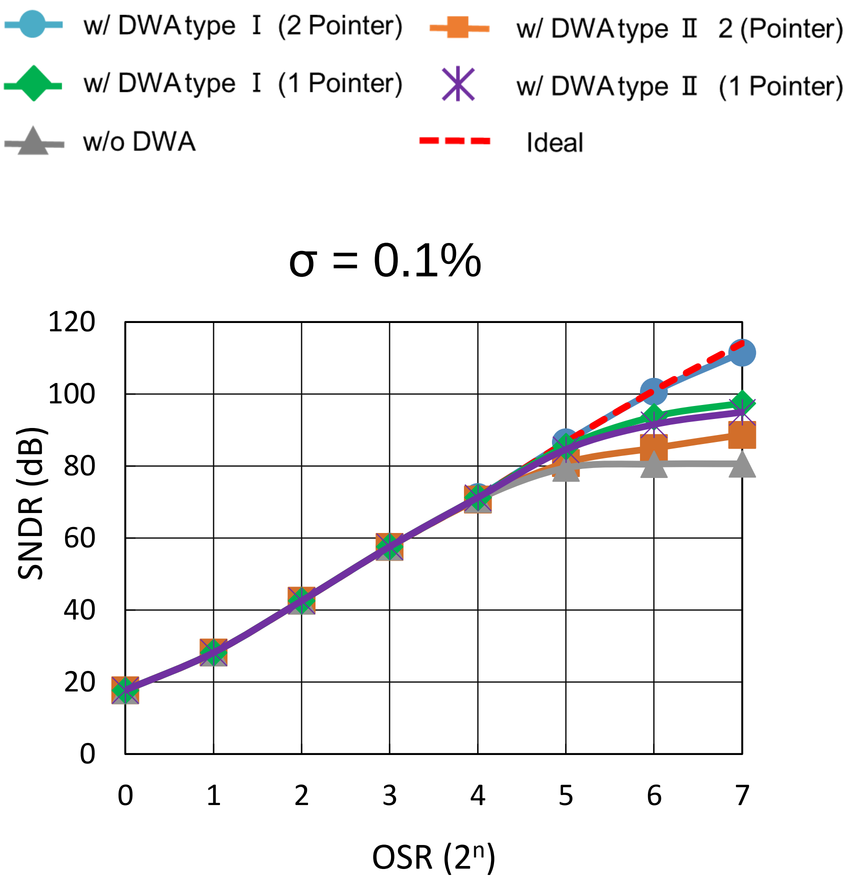


Good Algorithm

・noise-shaping



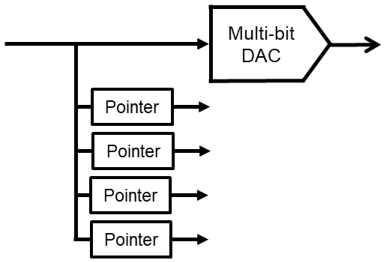
< DWA type I >

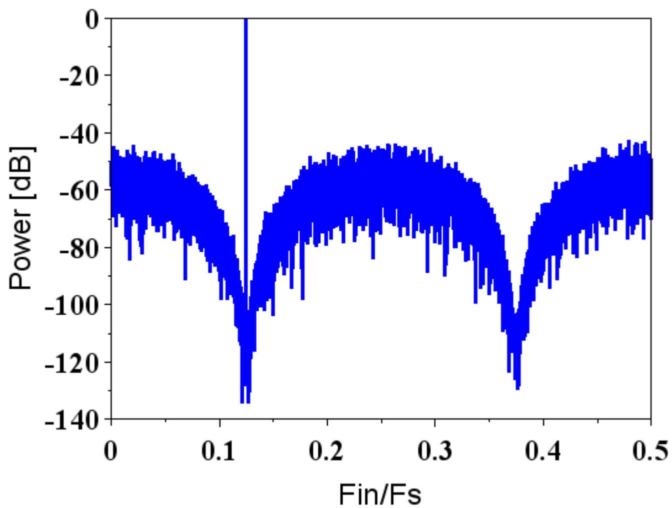


Band-Pass ΔΣDAC (Ternary)

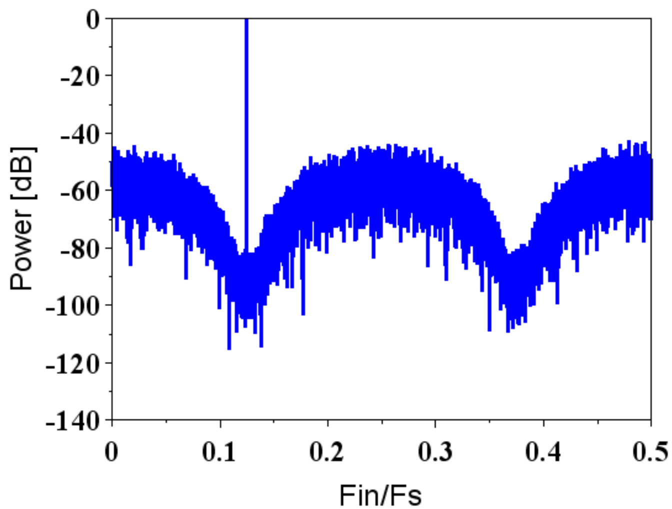
- Band-pass (BP) ΔΣ DAC (N=4)

  Segmented DAC with **ternary** unit cells





< DWA type I >

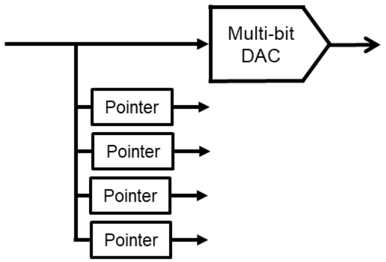


< DWA type II >

Band-Pass ΔΣDAC (Ternary)

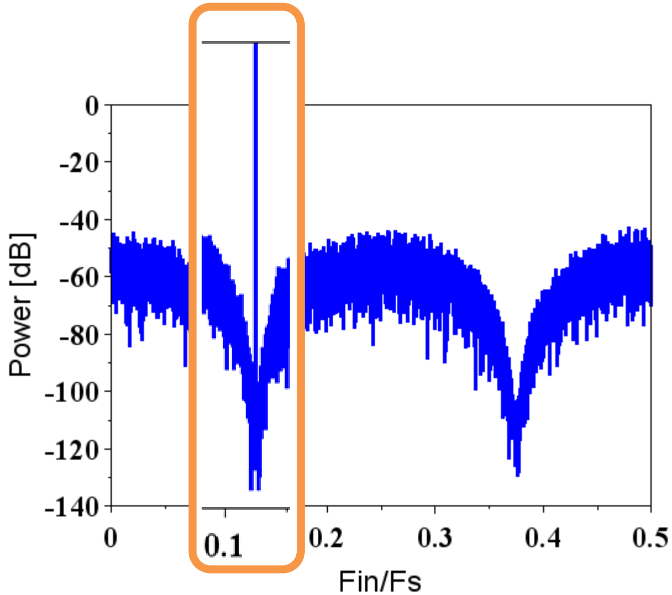
- Band-pass (BP) ΔΣ DAC (N=4)

  Segmented DAC with ternary unit cells

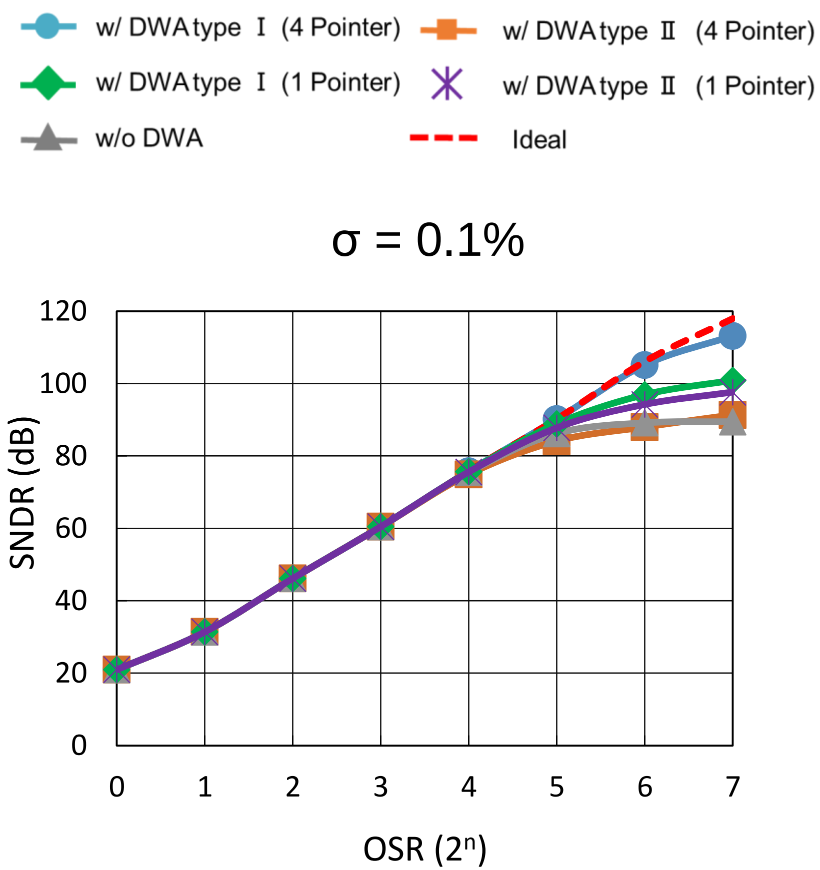


Good Algorithm

・noise-shaping



< DWA type I >



Outline

- Research Background
- ΔΣDA Converter
  - DWA*Algorithm (* Data-Weighted Averaging)
- Simulation verification
  - Binary, Ternary DWA Overview
  - ΔΣ DA Converter : HP type
  - ΔΣ DA Converter : BP type
- Conclusion

Conclusion

- HP, BP multi-bit ΔΣ DACs

- In case HP, BP ΔΣ DACs with **ternary** unit cells, DWA type I with pointers alternately used is effective.

Thank you for attention

Appendix

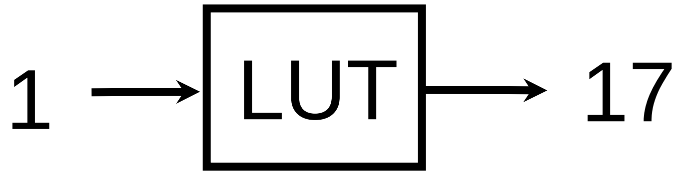
App.

Look Up Table (LUT)

LUT: Save data in advance ⇒ Data corresponding to input output

example

Cat's age	Equivalent human age
1	17
2	23
3	28
4	32
5	36
6	40
7	44
8	48



Cats one year old

Human being 17 years old

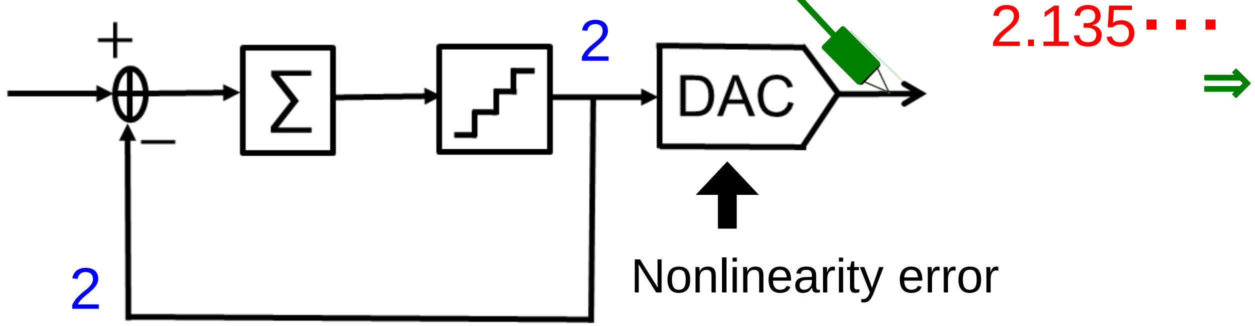
App.

Self-Calibration Algorithm

Preparation

⇒ Feed back value measured with high precision ΔΣ ADC

Save to LUT

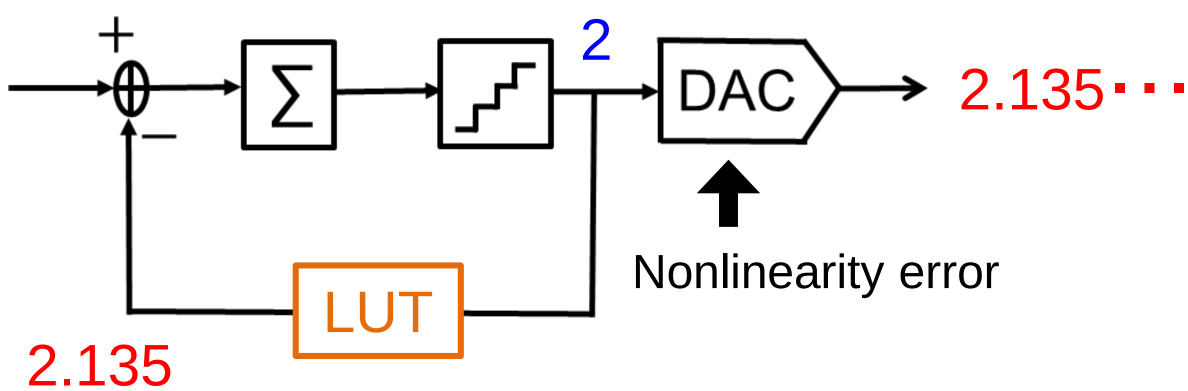


LUT

Input	Output
0	
1	
2	2.135
3	

⋮

Implementation of self-calibration algorithm



LUT

Input	Output
0	0.000
1	1.241
2	2.135
3	2.926

⋮