A Second-order DWA Algorithm for Multi-bit ΔΣ ADC/DAC

Hiroyuki Hagiwara, Yuanyang Du, Masahiro Murakami, Hao San, Anna Kuwana, Haruo Kobayashi

Gunma University
Kobayashi Laboratory

Tokyo City University

群馬大学
GUNMA UNIVERSITY

OUTLINE

Introduction & Objective

What is DWA ?

Multi-bit ΔΣADC/DAC

Proposed 2nd-order DWA Algorithm

Circuit Realization of 2nd-order DWA Algorithm

Simulation Verification

Summary

OUTLINE

Introduction & Objective

What is DWA ?

Multi-bit ΔΣADC/DAC

Proposed 2nd-order DWA Algorithm

Circuit Realization of 2nd-order DWA Algorithm

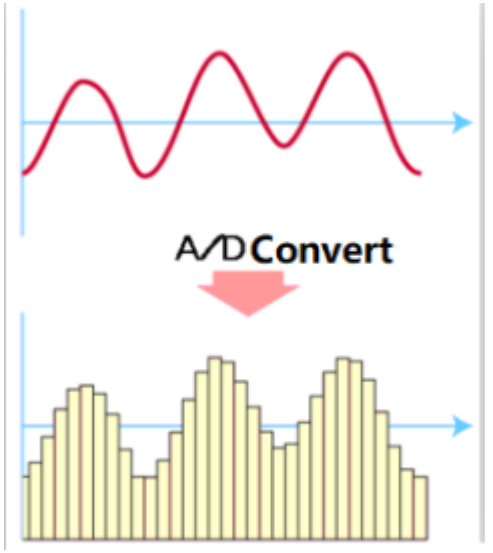
Simulation Verification

Summary

Background

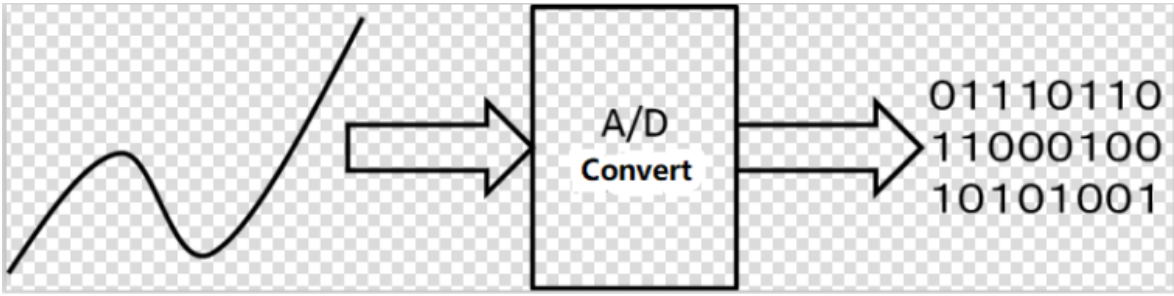
The ΔΣ ADCs/DACs are used for IoT applications → **Sensor Face Interface**
Narrow Band Internet of Things

Mult bit configuration
(a)small hardware
(b)non-linearity



DWA algorithm

- **conventional:1st-order DWA**
- **our approach:2nd-order DWA**



What is DWA ?

ADC/DAC/TDC digital calibration techniques prevail in nano-CMOS era.

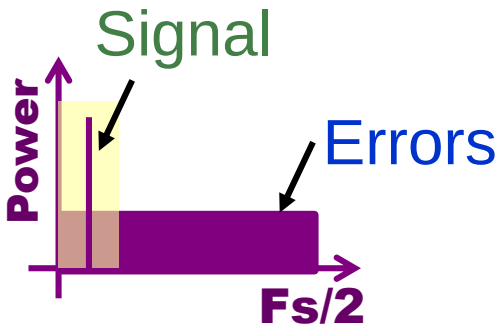
- Error Correction
  - No measurement of errors
  - Redundancy usage

DWA:
Data Weighted Averaging

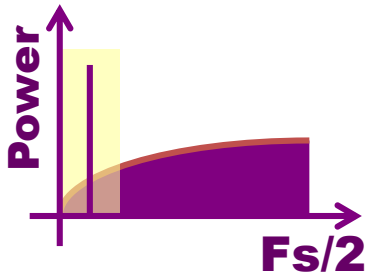
DWA Techniques

- Error Correction
  - No measurement of errors
  - Redundancy usage
    - Time averaging of errors
    - Spectrum shaping of errors

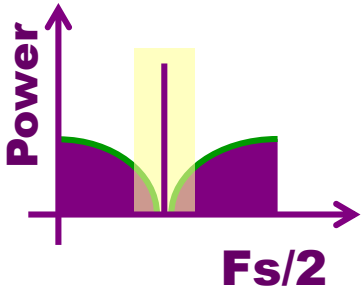
DWA:
Data Weighted Averaging
DEM:
Dynamic Element Matching



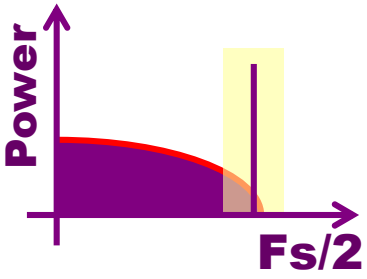
W/O DWA



LP DWA



BP DWA

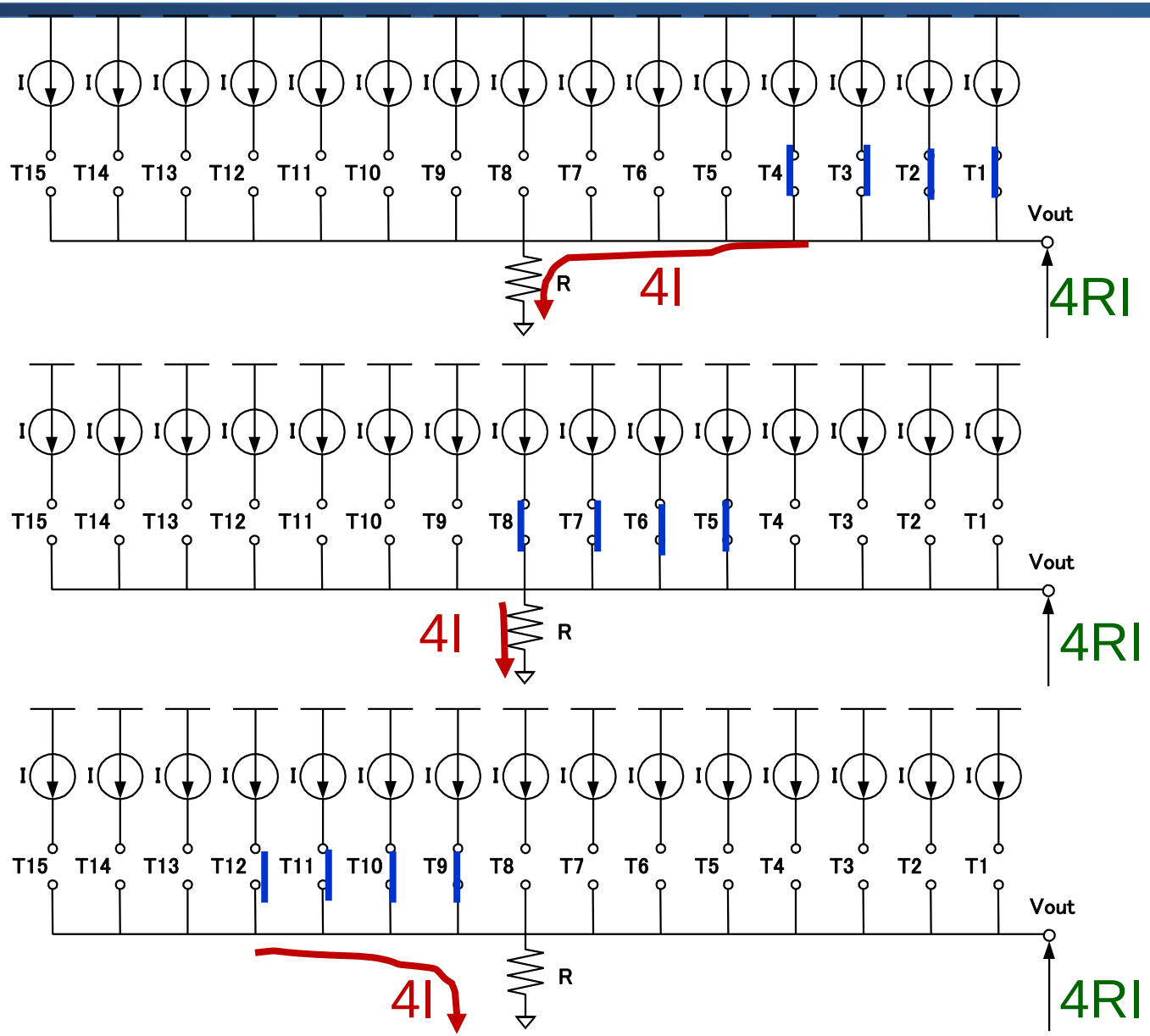


HP DWA

Segment DAC with Redundancy

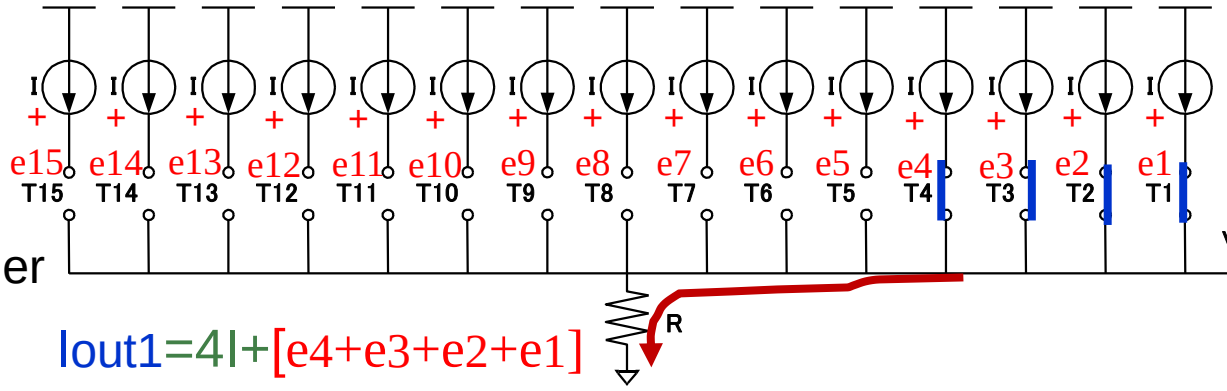
Digital input = 4

Multiple realization configurations



Unit Cell Mismatches

Current mismatches e15, e14,…,e1 spectrum shaping by cell selection order

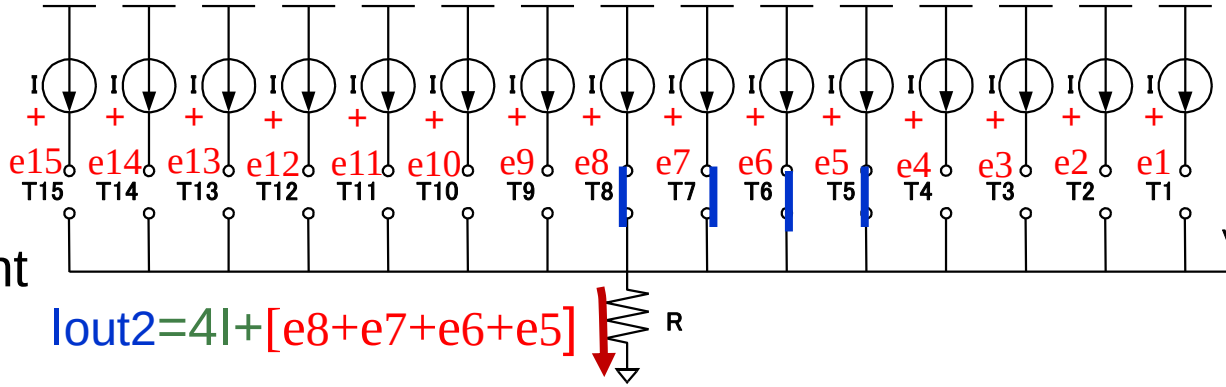


Iout1=4I+[e4+e3+e2+e1]

4RIout1

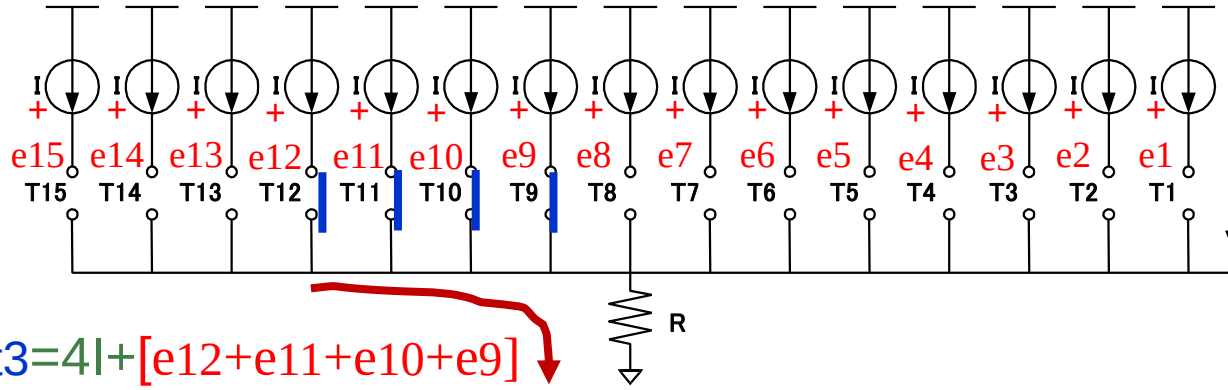
DWA algorithm

No measurement of e15, e14,…,e1



Iout2=4I+[e8+e7+e6+e5]

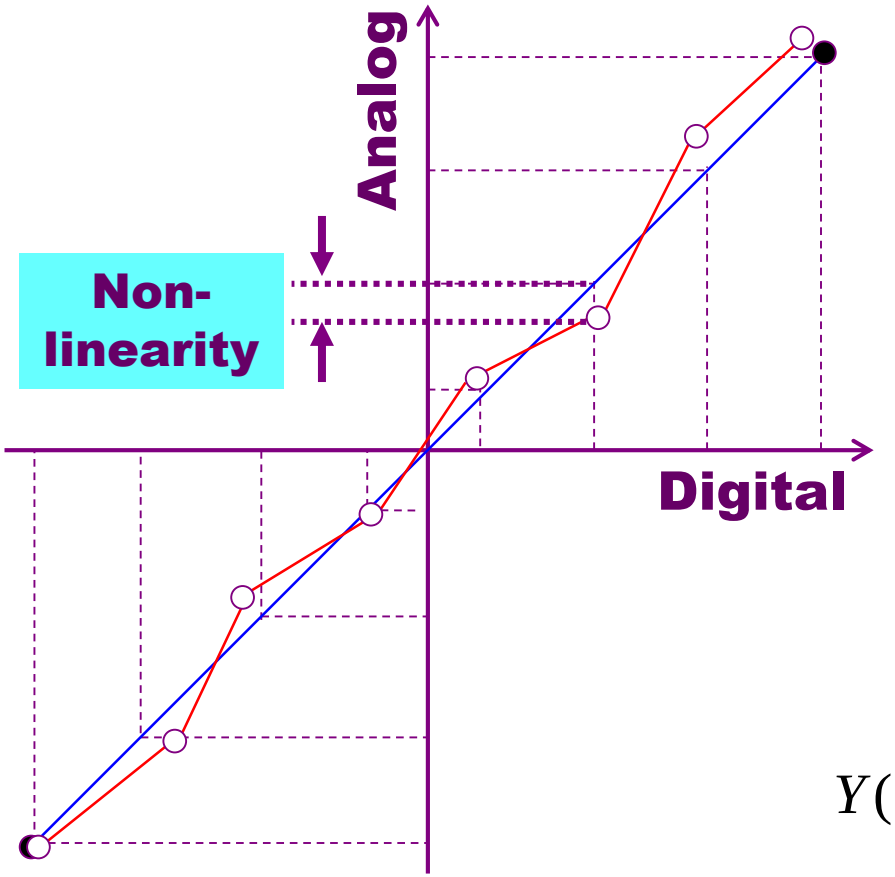
4RIout2



Iout3=4I+[e12+e11+e10+e9]

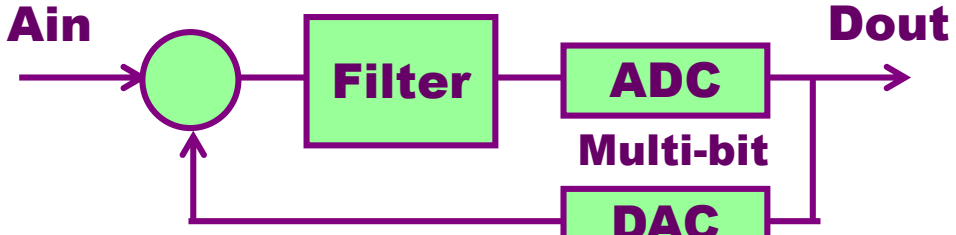
4RIout3

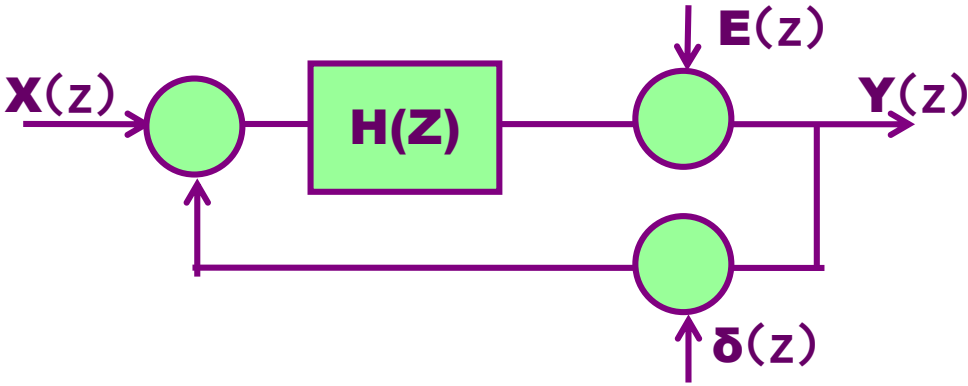
Multi-bit DAC Nonlinearity



● :Single-bit Output

○ :Multi-bit Output



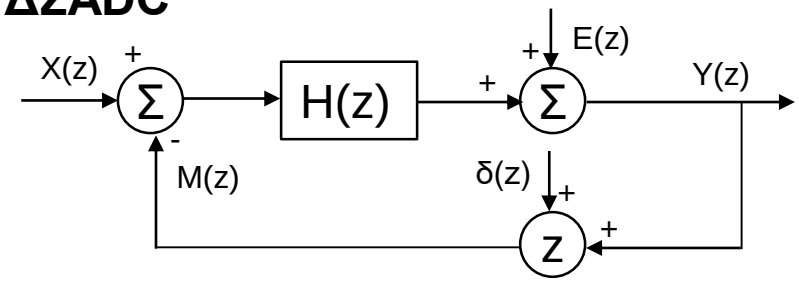


$$Y(z) = \frac{H(z)}{1+H(z)}\{X(z) - \delta(z)\} + \frac{1}{1+H(z)}E(z)$$

δ(z) is NOT noise-shaped

Multi-bit ΔΣADC/DAC

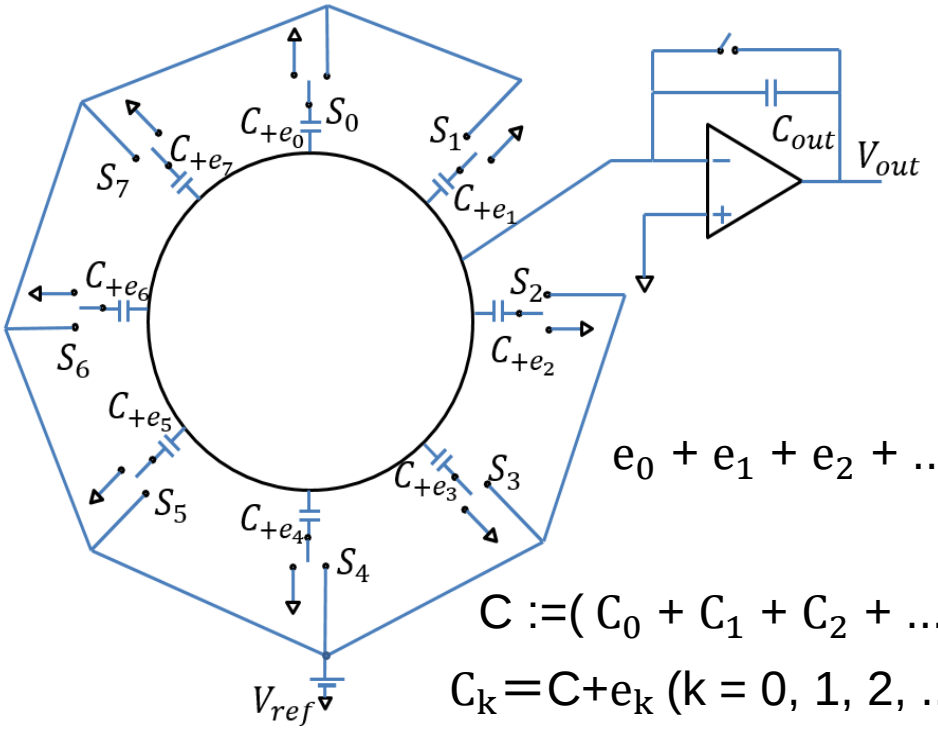
ΔΣADC



$$Y(z) = \frac{H(z)}{1+H(z)}[X(z) - \delta(z)] + \frac{1}{1+H(z)}E(z)$$

nonlinearity

a multi-bit(9bit) DAC with the switched capacitor



When a digital input is *m*
and the DWA algorithm is not used,
then the output voltage

$$V_{out} = -m\frac{C}{C_{out}}V_{ref} + \delta$$

$$e_0 + e_1 + e_2 + \ldots + e_7 = 0$$

Nonlinearity δ of the DAC is expressed by

$$\delta = -\frac{e_0+e_1+e_2+\ldots+e_{m-1}}{C_{out}}V_{ref}$$

$$C := (C_0 + C_1 + C_2 + \ldots + C_7)/8$$

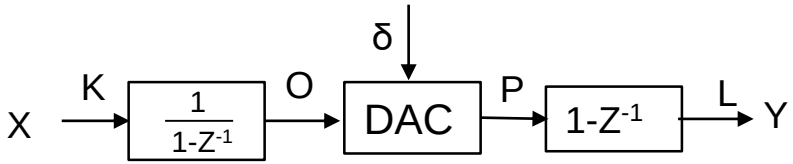
$$C_k = C + e_k \ (k = 0, 1, 2, \ldots, 7).$$

OUTLINE

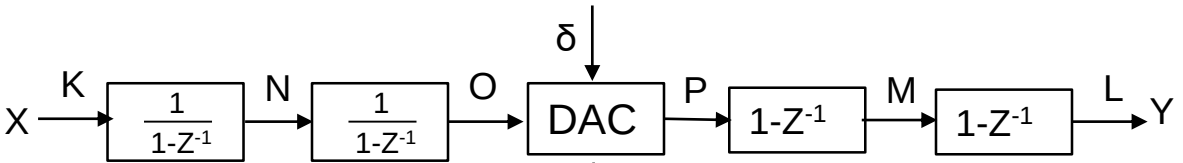
Proposed 2nd-order DWA Algorithm

2nd-order Noise-Shaping of DAC Nonlinearity

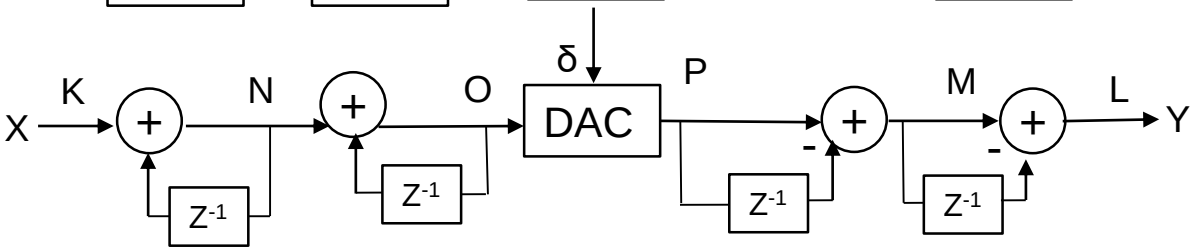
(a) 1st-order DWA architecture



(b) 2nd-order DWA architecture

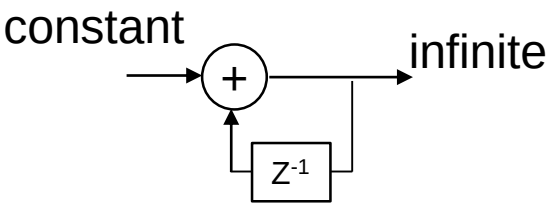


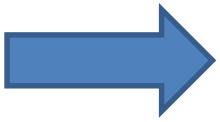
(c) equivalent block diagram



$$Y(z) = X(z) + (1 - z^{-1})^2 \delta(z)$$

problem





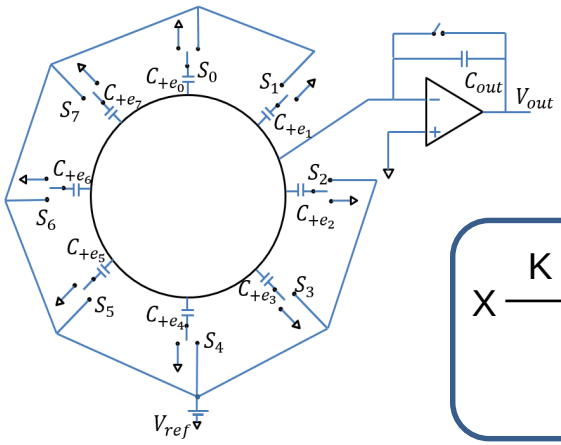
does not work well

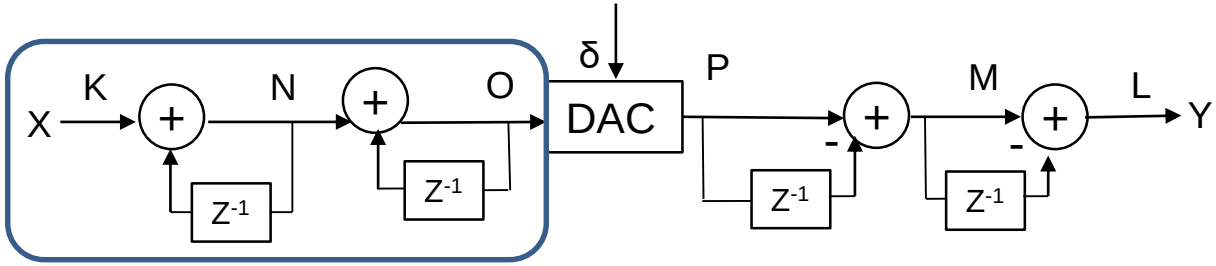
Operation Proposed 2nd-order DWA Algorithm(1)

Example

DAC input at time 0: D(0)=3

Input	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
3	+	+	+					





Operation Proposed 2nd-order DWA Algorithm(2.1)

A. Two pointers Po+, Po-

Plus side:

Pointer: $P_{0+}(n)=\mathrm{mod}_8[D(n-1)+\sum_{k=0}^{n-2}P_{0+}(k)]$

Signal start: $S_+(n)=\mathrm{mod}_8[P_{0+}(n)+1]$

Number: $A_+(n)=D(n)+\mathrm{mod}_8[A_+(n-1)]$

Minus side:

Pointer: $P_{0-}(n)=P_{0+}(n-1)$

Signal start: $S_-(n)=\mathrm{mod}_8[P_{0+}(n-1)+1]$

Number: $A_-(n)=\mathrm{mod}_8[A_+(n-1)]$

Example

DAC input at time 1: D(1)=4

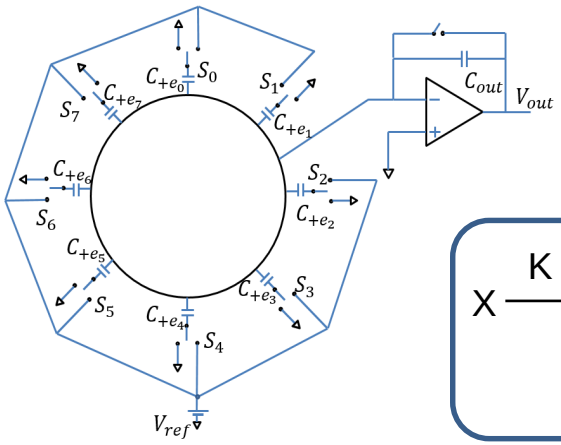
$P_{0+}(1)=3$ $P_{0-}(1)=0$

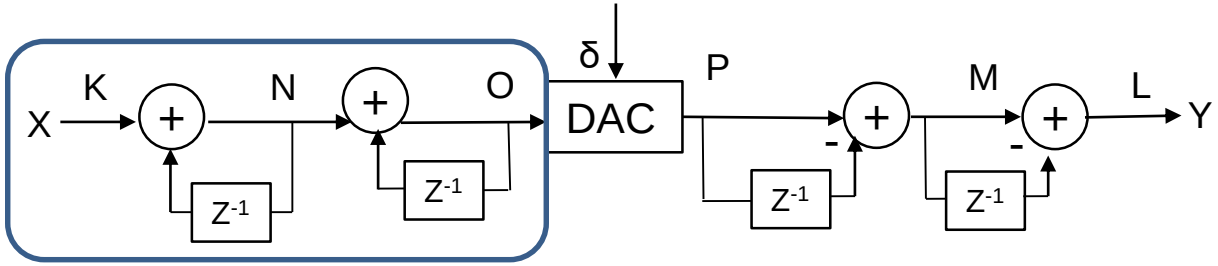
$S_+(1)=4$ $S_-(1)=1$

$A_+(1)=7$ $A_-(1)=3$

Input	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
4				+	+	+	+	+
	+	+						

D(0)+D(1)=7





Operation Proposed 2nd-order DWA Algorithm(2.2)

B. Assignment of "Positive" and "Negative" to Cell

(i) Assign "positive : +" to the unit cells of

- $S_+(n)$
- $\mathrm{mod}_8(S_+(n)+1)$
- $\mathrm{mod}_8(S_+(n)+2)$
- …
- $\mathrm{mod}_8(S_+(n)+A_+-1)$

(ii) Assign " negative : -" to the unit cells of

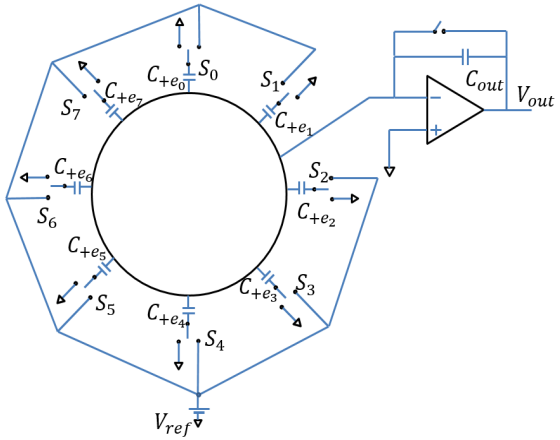
- $S_-(n)$
- $\mathrm{mod}_8(S_-(n)+1)$,
- $\mathrm{mod}_8(S_-(n)+2)$,
- ….
- $\mathrm{mod}_8(S_-(n)+A_--1)$

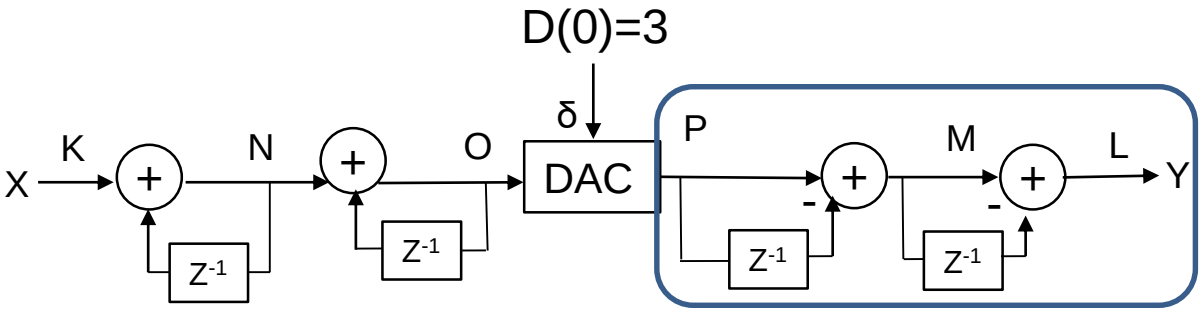
Example

DAC input at time 1: D(1)=4

$P_{0+}(1)=3$ $P_{0-}(1)=0$
$S_+(1)=4$ $S_-(1)=1$
$A_+(1)=7$ $A_-(1)=3$

input	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
4				+	+	+	+	+
	+	+						
	-	-	-					





Operation Proposed 2nd-order DWA Algorithm(2.3)

C. Decision of Unit Cell Value: -1, 0, 1 or 2

(i)Condition for unit cell value of -1:

"+ is k+2 times assigned" and "– is k+3 times assigned."

(ii) Condition for unit cell value of 0:

"+ is k+2 times assigned" and "– is k+2 times assigned."

(iii)Condition for unit cell value of +1:

"+ is k+2 times assigned" and "– is k+1 times assigned."

(iv) Condition for unit cell value of 2:

"+ is k+2 times assigned" and "– is k times assigned."

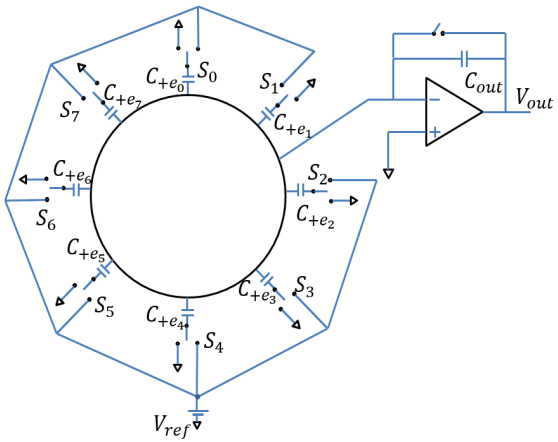
Example

DAC input at time 1: D(1)=4

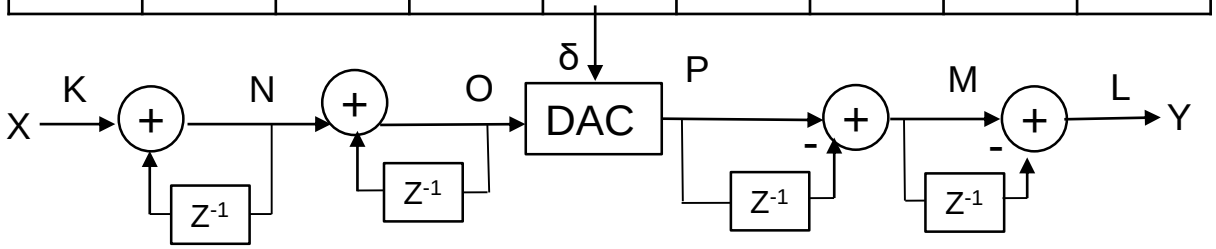
$P_{0+}(1)=3$ $P_{0-}(1)=0$

$S_{+}(1)=4$ $S_{-}(1)=1$

$A_{+}(1)=7$ $A_{-}(1)=3$



input	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
4				+	+	+	+	+
	+	+						
	-	-	-					
result			-	+	+	+	+	+



Operation Proposed 2nd-order DWA Algorithm(3.1)

A. Two pointers Po+, Po-

Plus side:

Pointer: $P_{0+}(n)=\mathrm{mod}_8[D(n-1)+\sum_{k=0}^{n-2}P_{0+}(k)]$

Signal start: $S_+(n)=\mathrm{mod}_8[P_{0+}(n)+1]$

Number: $A_+(n)=D(n)+\mathrm{mod}_8[A_+(n-1)]$

Minus side:

Pointer: $P_{0-}(n)=P_{0+}(n-1)$

Signal start: $S_-(n)=\mathrm{mod}_8[P_{0+}(n-1)+1]$

Number: $A_-(n)=\mathrm{mod}_8[A_+(n-1)]$

Example

DAC input at time 2: D(2)=2

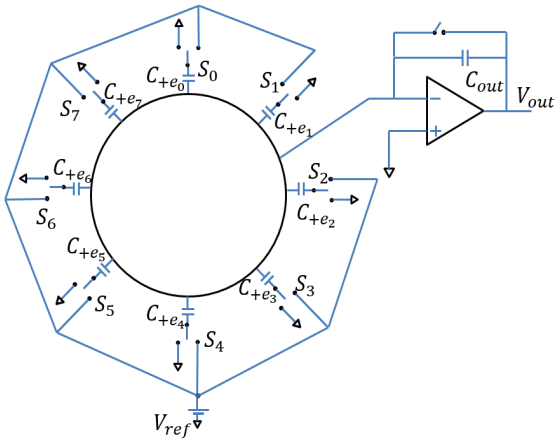
$P_{0+}(2)=4$ $P_{0-}(2)=3$

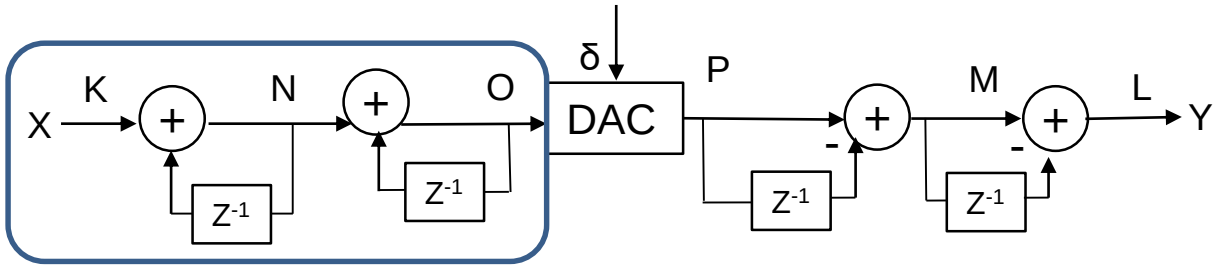
$S_+(2)=5$ $S_-(2)=4$

$A_+(2)=9$ $A_-(2)=7$

Input	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
2			+	+	+	+	+	+
	+	+	+					

D(0)+D(1)+D(2)=9





Operation Proposed 2nd-order DWA Algorithm(3.2)

B. Assignment of "Positive" and "Negative" to Cell

(i) Assign "positive : +" to the unit cells of

- $S_+(n)$
- $\mathrm{mod}_8(S_+(n)+1)$
- $\mathrm{mod}_8(S_+(n)+2)$
- …
- $\mathrm{mod}_8(S_+(n)+A_+-1)$

(ii) Assign " negative : -" to the unit cells of

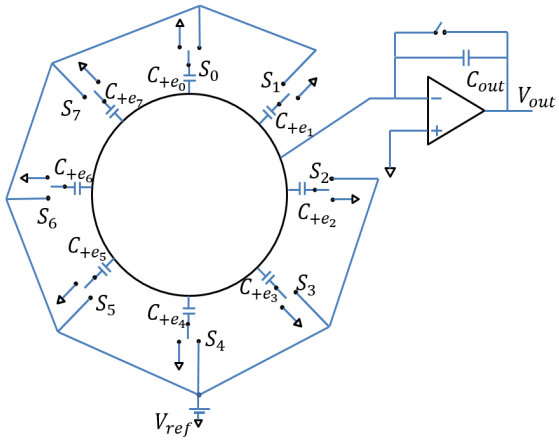
- $S\_(n)$
- $\mathrm{mod}_8(S\_(n)+1)$,
- $\mathrm{mod}_8(S\_(n)+2)$,
- ….
- $\mathrm{mod}_8(S\_(n)+A\_-1)$

Example

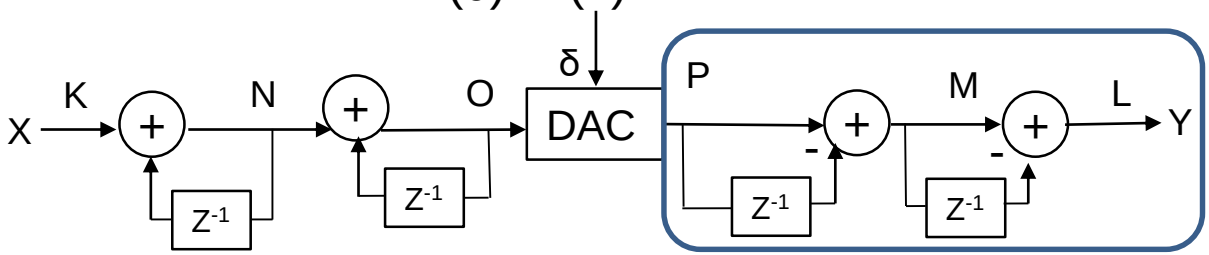
DAC input at time 2: D(2)=2

$P_{0+}(2)=4$ $P_{0-}(2)=3$
$S_+(2)=5$ $S_-(2)=4$
$A_+(2)=9$ $A_-(2)=7$

Input	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
2			+	+	+	+	+	+
	+	+	+					
				-	-	-	-	-
	-	-						



D(0)+D(1)=7



Operation Proposed 2nd-order DWA Algorithm(3.3)

C. Decision of Unit Cell Value: -1, 0, 1 or 2

(i)Condition for unit cell value of -1:

"+ is k+2 times assigned" and "– is k+3 times assigned."

(ii) Condition for unit cell value of 0:

"+ is k+2 times assigned" and "– is k+2 times assigned."

(iii)Condition for unit cell value of +1:

"+ is k+2 times assigned" and "– is k+1 times assigned."

(iv) Condition for unit cell value of 2:

"+ is k+2 times assigned" and "– is k times assigned."

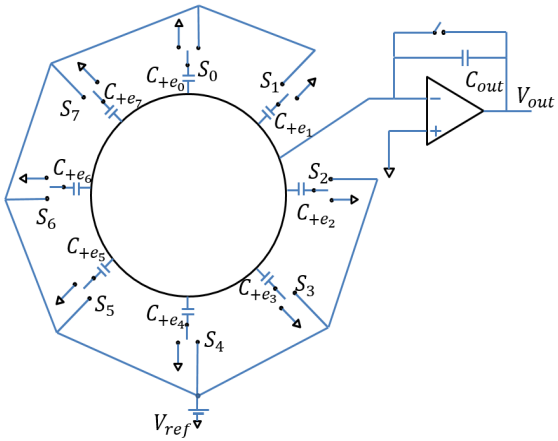
Example

DAC input at time 2: D(2)=2

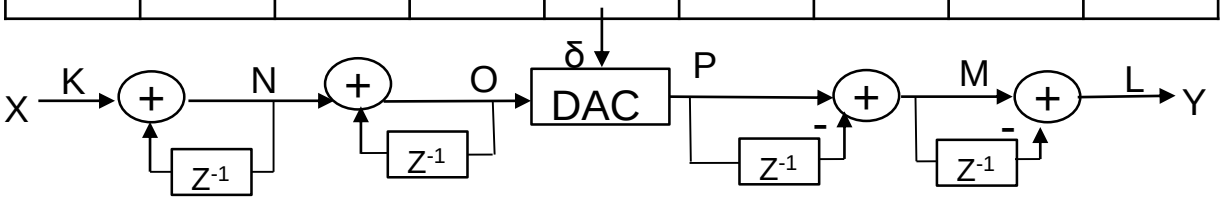
$P_{0+}(2)=4$ $P_{0-}(2)=3$

$S_{+}(2)=5$ $S_{-}(2)=4$

$A_{+}(2)=1$ $A_{-}(2)=7$

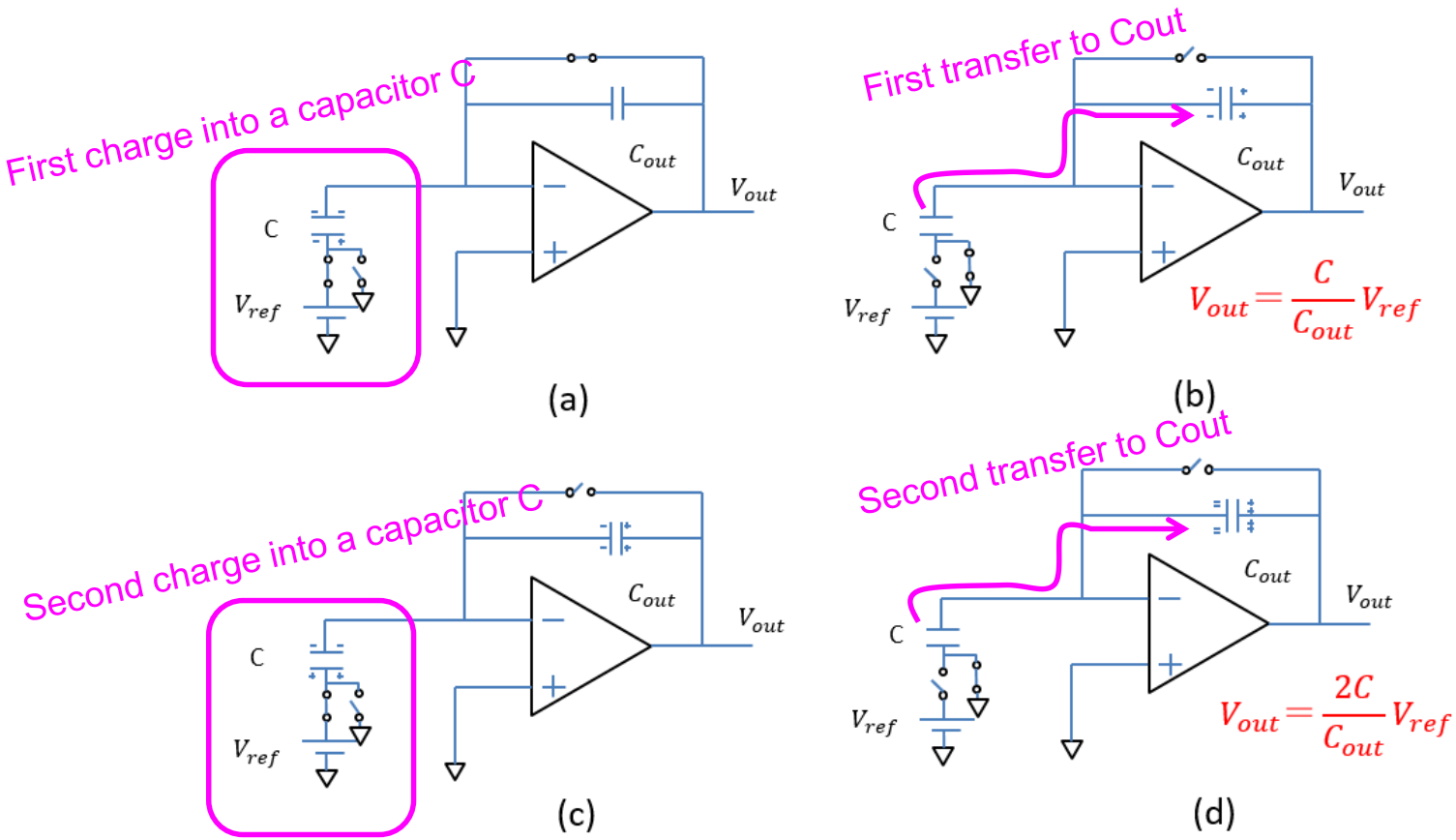


Input	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
2			+	+	+	+	+	+
	+	+	+					
				-	-	-	-	-
	-	-						
result			++					



Circuit Realization of 2nd-order DWA Algorithm

DWA algorithm → needs → twice unit cell output and minus unit cell output → realized with 2 clock operation



Simulation Verification

Matlab simulations

ideal DAC
(without mismatches)

with mismatches
without DWA

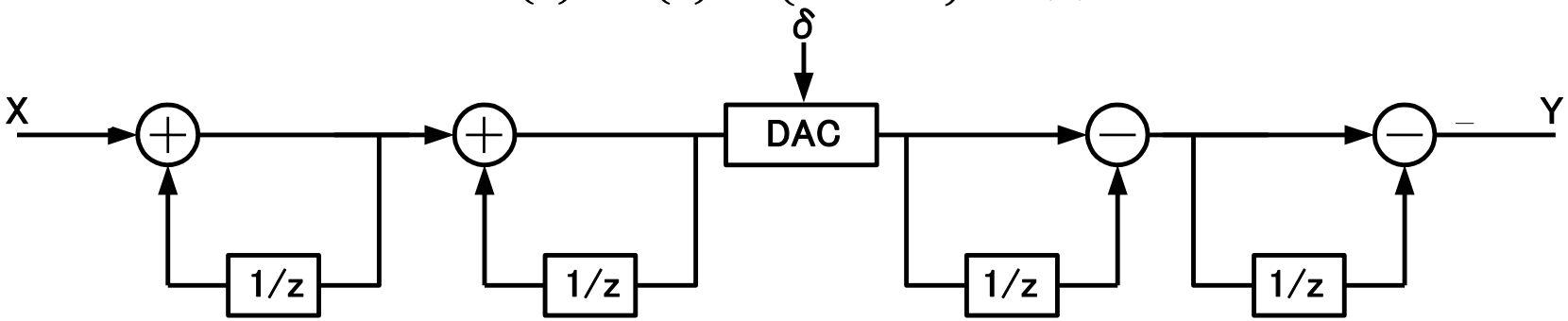
mismatches
with the 1st-order DWA

with mismatches
with the proposed
2nd-order DWA

2^{nd}-order DWA algorithm is effective

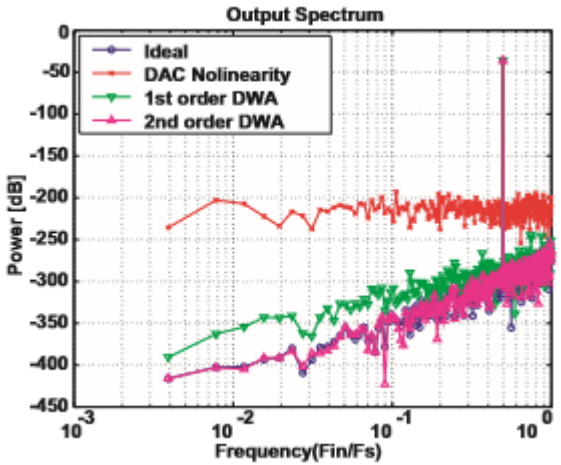
2nd-order DWA

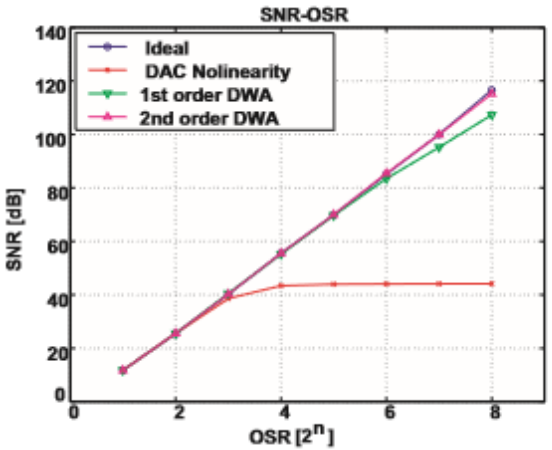
$$Y(z) = X(z) + (1 - z^{-1})^2 \delta(z)$$



2 integrators

2 differentiators





- 2nd-order DWA is more effective
- But its circuit/operation become complicated

Summary

- a second-order DWA algorithm
- reduce mismatches with 2nd-order noise-shaping
- requires a two clock operation with switched capacitor circuit
- make the circuit operation slow, but easy to implement

thank you for your listening