SAR ADC Algorithm with Redundancy Based on Fibonacci Sequence

Yutaro Kobayashi **Haruo Kobayashi**

Gunma University

Supported by ***STARC***

Kobayashi Laboratory

群馬大学 GUNMA UNIVERSITY

Research Objective

Objective

- **Development of**
  Reliable & High-speed SAR ADC

Our Approach

- **Redundant Search Algorithm Design with Number Theory**

Outline

- Research Background
- SAR ADC Redundancy Design
- Proposed SAR Algorithm Using Fibonacci Sequence
- Advantages of Proposed SAR Algorithm
- Conclusions

Outline

- **Research Background**
- SAR ADC Redundancy Design
- Proposed SAR Algorithm Using Fibonacci Sequence
- Advantages of Proposed SAR Algorithm
- Conclusions

Research Background

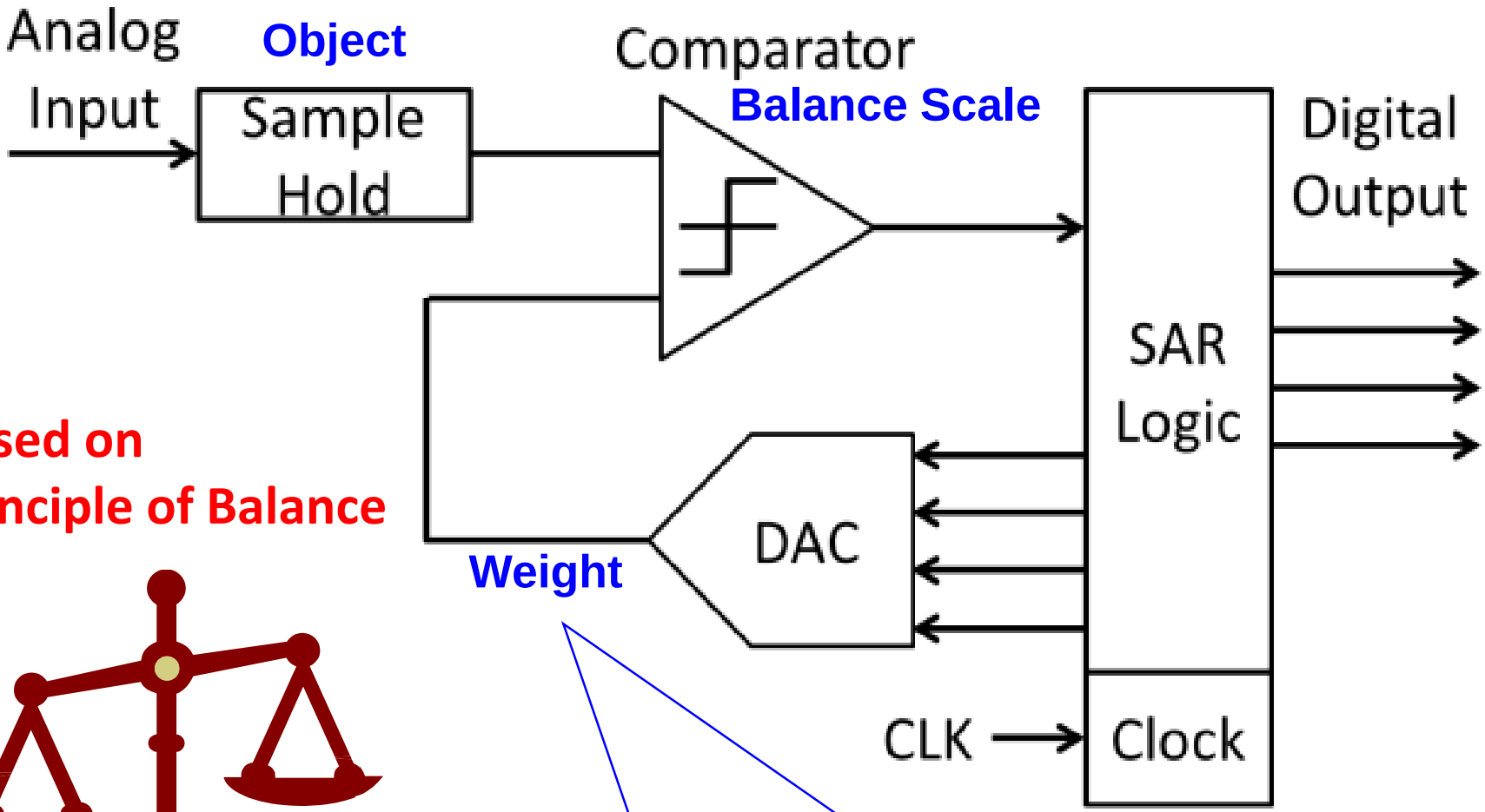
Automotive Electronics are in spotlight

High-speed, reliable
“SAR ADC” in microcontroller is needed

Redundancy design for error correction

Design issues

SAR ADC Configuration



Based on Principle of Balance

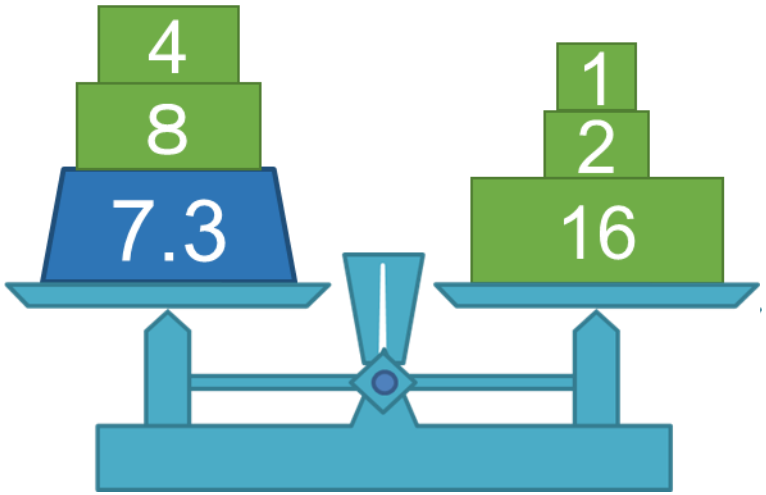
Generally use binary weight

(1 , 2 , 4 , 8 , 16 , 32, 64 …)

Binary Search SAR ADC Operation(1)

5bit5step SAR ADC

- **Binary Weight**
  p(k) = 16, 8, 4, 2, 1
- **input 7.3 [LSB]**



7.3 ⇒ 7

Step		1st	2nd	3rd	4th	5th	output
Weight p(k)		16	8	4	2	1	
Level	31						31
	30						30
	29						29
	28						28
	27						27
	26						26
	25						25
	24						24
	23						23
	22						22
	21						21
	20						20
	19						19
	18						18
	17						17
	16						16
	15						15
	14						14
	13						13
	12						12
	11						11
	10						10
	9						9
	8						8
	7						7
	6						6
	5						5
	4						4
	3						3
	2	0	0	1	1	1	2
	1						1
	0						0

Binary Search SAR ADC Operation(2)

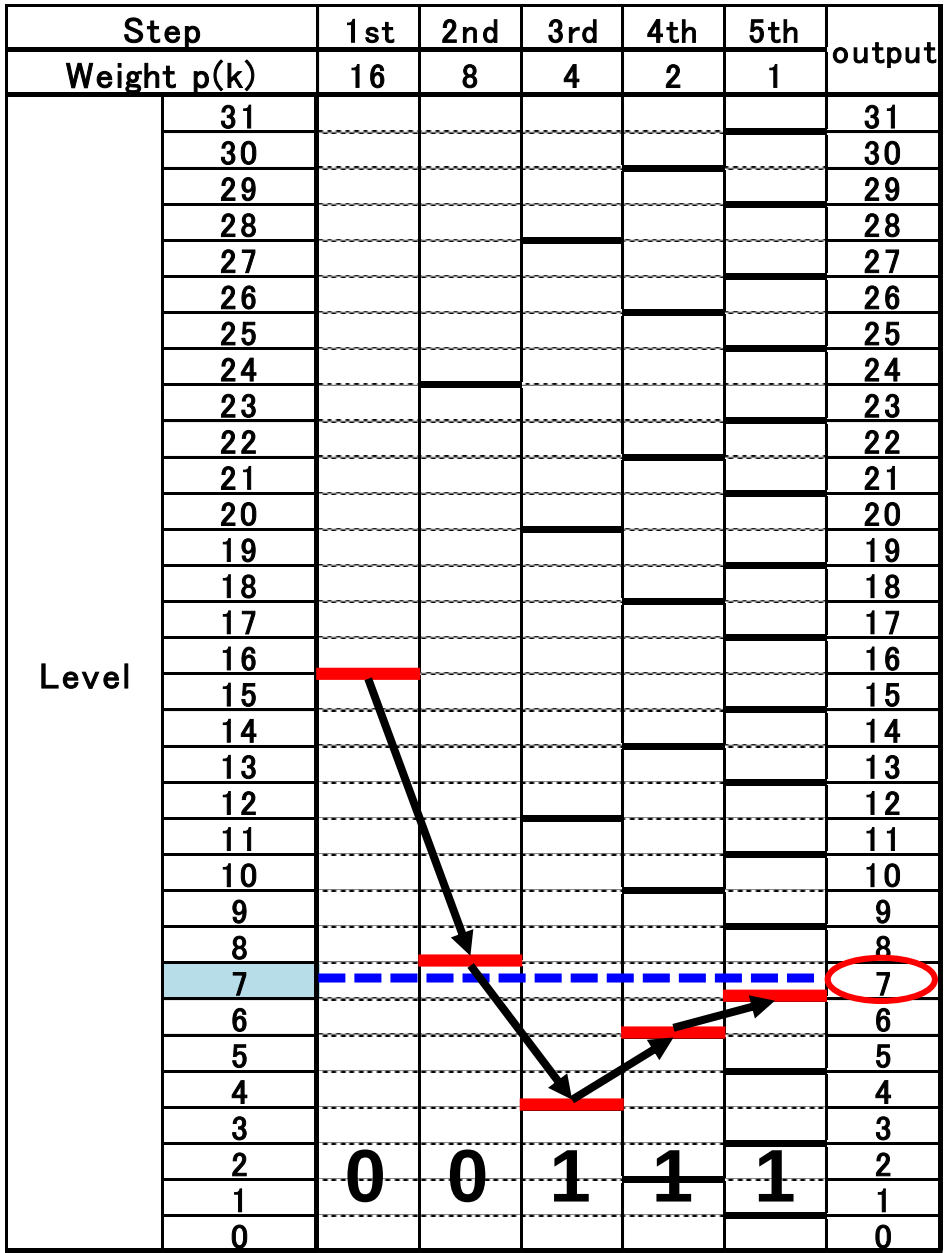
5bit5step SAR ADC

- **Binary Weight**
  p(k) = 16, 8, 4, 2, 1
- **input 7.3 [LSB]**

One-to-one mapping
between Decimal and Binary codes

$$D_{out} = (00111)_2$$

7=16− 8− 4+2+1+ 0.5-0.5



Binary Search SAR ADC Operation(3)

5bit5step SAR ADC

- Binary Weight
  p(k) = 16, 8, 4, 2, 1
- input 7.3 [LSB]

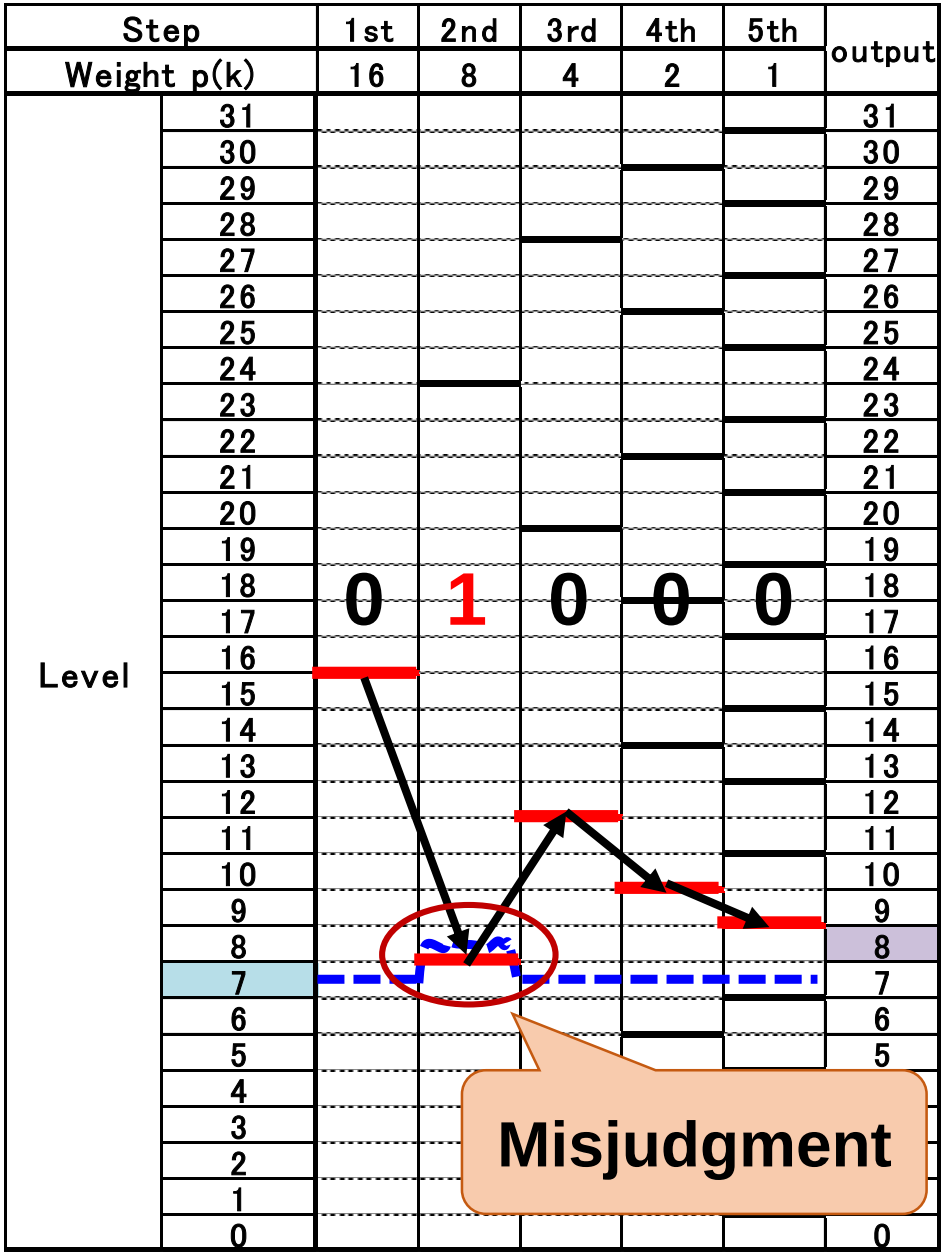
One-to-one mapping
between Decimal and Binary codes

D_{out} =(00111)$_2$=7

1 Misjudgment leads to
incorrect output

D_{out} =(01000)$_2$=8

Reliability problem !



Outline

- Research Background
- **SAR ADC Redundancy Design**
- Proposed SAR Algorithm Using Fibonacci Sequence
- Advantages of Proposed SAR Algorithm
- Conclusions

SAR ADC Redundancy Design

Redundancy: Surplus, Extra

Apply to SAR ADC

Using Time Redundancy

- **Increase comparison steps**
- **Change reference voltages**

Multiple output expressions

Enable Digital error correction!

Binary weight
p(k): 1,2,4,8,16

Non-Binary weight
p(k): 1,2,3,6,10,16

Redundant Search SAR ADC Operation(1)

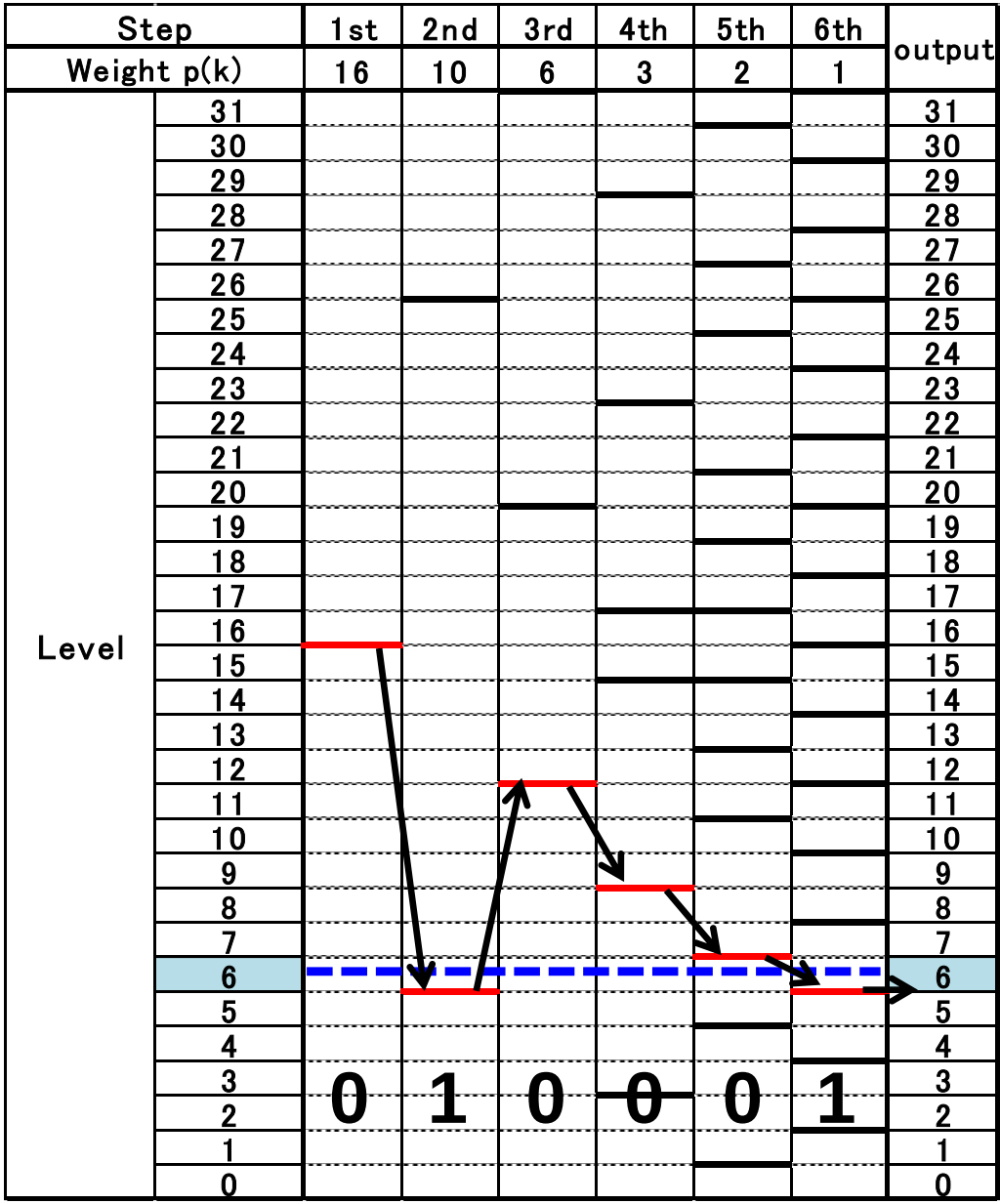
5bit6step SAR ADC

- **Redundant Weight**
  p(k) = 16, 10, 6, 3, 2, 1
- **input 6.3 [LSB]**

Increase number of comparison steps

6⇒010001⇒6

16−10+6−3−2−1+0.5-0.5
= 6



Redundant Search SAR ADC Operation(2)

5bit6step SAR ADC

- **Redundant Weight**
  p(k) = 16, 10, 6, 3, 2, 1
- **input 6.3 [LSB]**

Increase number of comparison steps

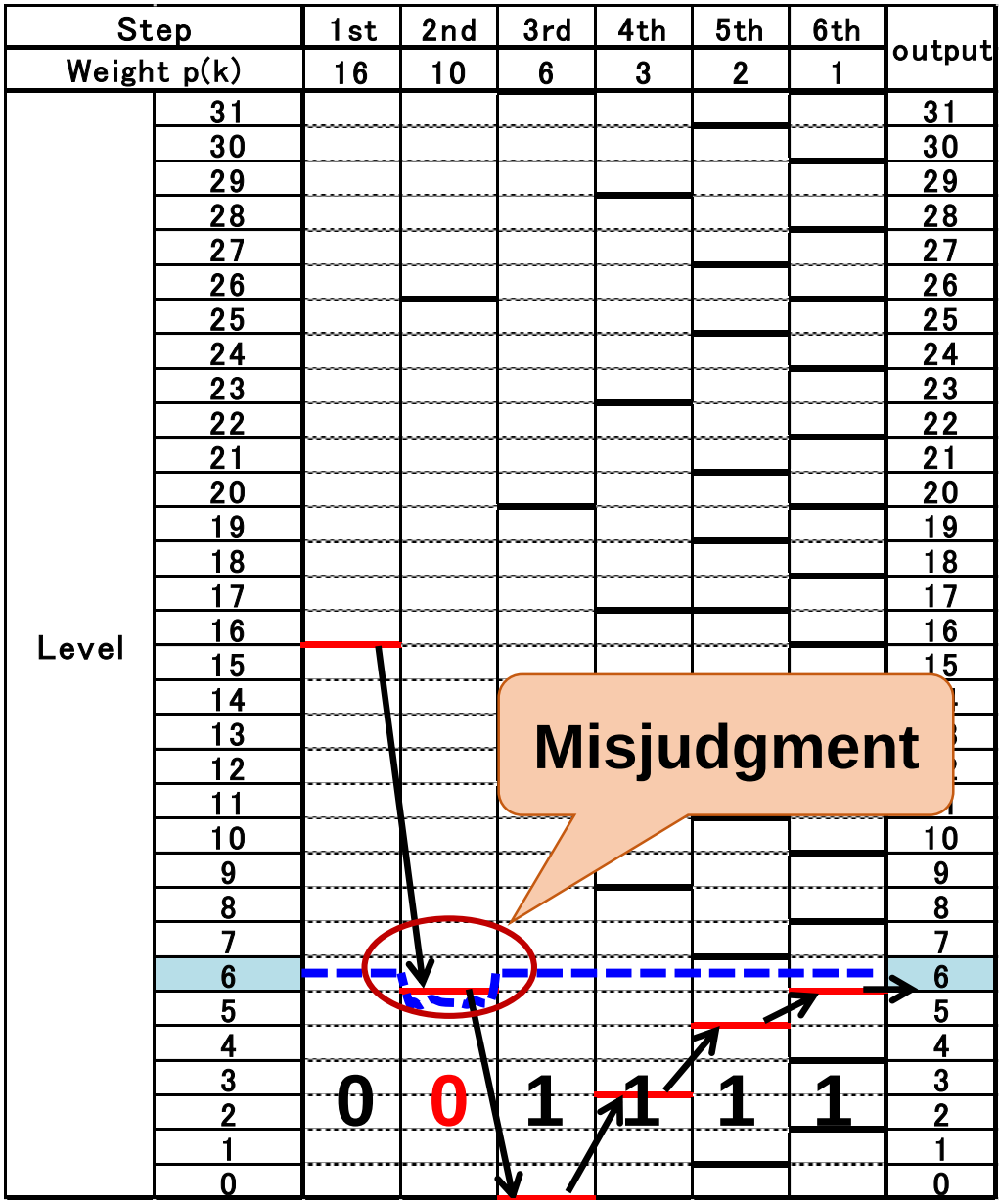
6⇒010001⇒6

Another expression

6⇒001111⇒6

Error correction

High-Reliability



Evaluation of Redundancy Design

5bit6step SAR ADC

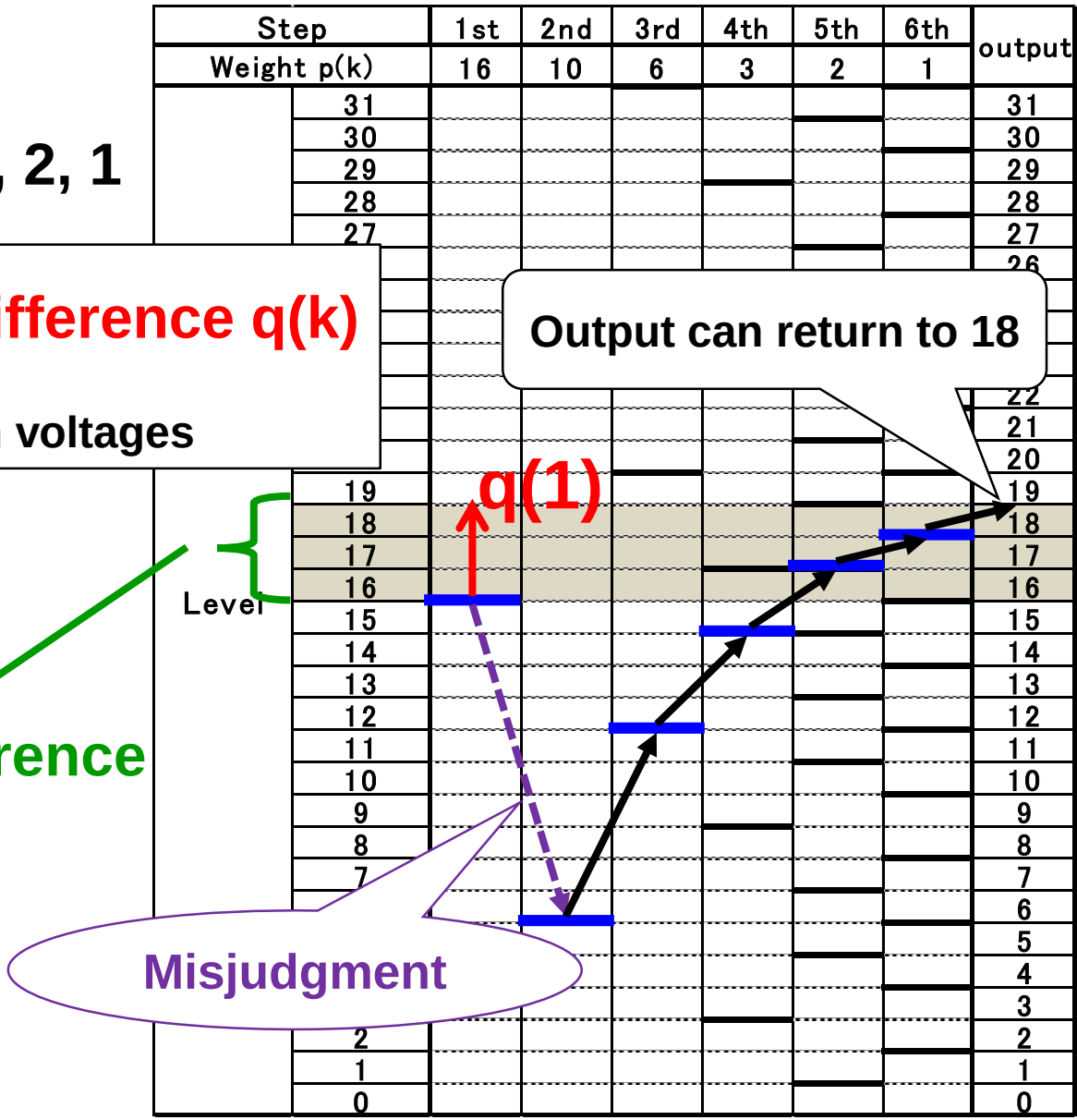
- **Redundant Weight**
  p(k) = 16, 10, 6, 3, 2, 1

Definition

k-th step correctable difference q(k)

q(k) is correctable difference between input and comparison voltages

1st step correctable difference
q(1)=3



Correctable Difference q(k)

5bit6step SAR ADC

- **Redundant Weight**
  p(k) = 16, 10, 6, 3, 2, 1

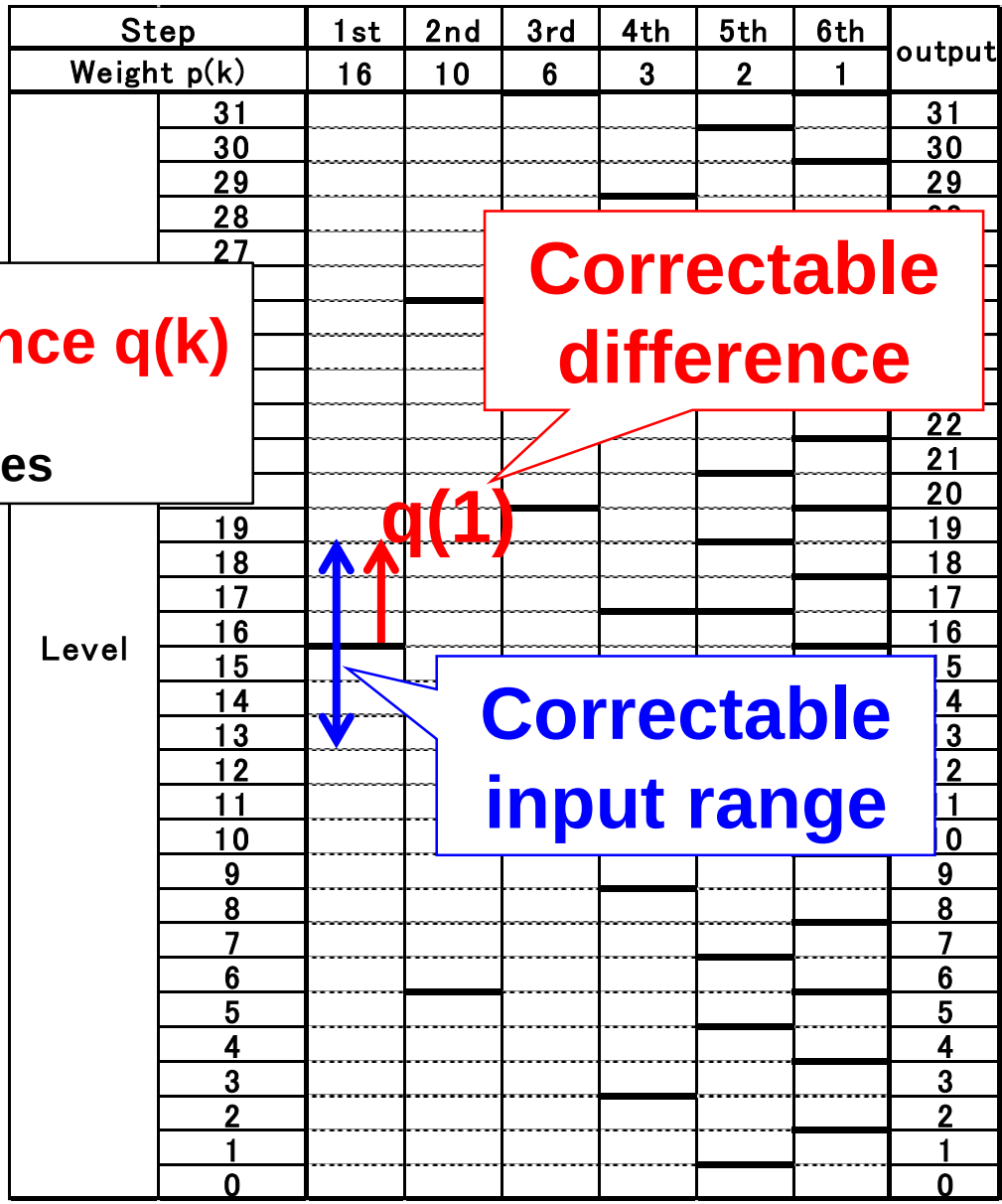
Definition

k-th step correctable difference q(k)
q(k) is correctable difference between input and comparison voltages

Condition of correctable

$$|Vin - Vcom(k)| \leq q(k)$$

Large q(k) indicates High-Reliability



Correctable Difference q(k)

The formula for q(k)

$$\mathbf{q(k) = -p(k+1) + 1 + \sum_{i=k+2}^{M} p(i)}$$

p(k) : reference voltage weight at k-th step
M : number of total step

q(k) determined only by p(k) !

Select proper p(k)

Step		1st	2nd	3rd	4th	5th	6th	output
Weight p(k)		16	10	6	3	2	1	
Level	31							31
	30							30
	29							29
	28							28
	27							27
	26			q(2)				26
	25							25
	24							24
	23							23
	22							22
	21							21
	20			q(3)				20
	19							19
	18							18
	17		q(1)					17
	16							16
	15							15
	14							14
	13							13
	12							12
	11							11
	10							10
	9							9
	8							8
	7							7
	6							6
	5							5
	4							4
	3							3
	2							2
	1							1
	0							0

Conventional Method of p(k) Selection

We select N bit and M step SAR ADC k-th step reference voltage p(k).
here $p(1) = 2^{N-1}$

Conventional method

Radix method

$$p(k) = r^{M-k} \text{ (here } 1 \leq r < 2\text{)}$$

Problems

- Difficult to select good radix
  - Trade-off of conversion steps and correction capability
- p(k) must be fraction
  - Fraction rate area causes low-precision
  - Rounding to integer causes dispersion of q(k)

Issues of Conventional Method

5bit6step SAR ADC

- **Radix method**

Radix=1.8

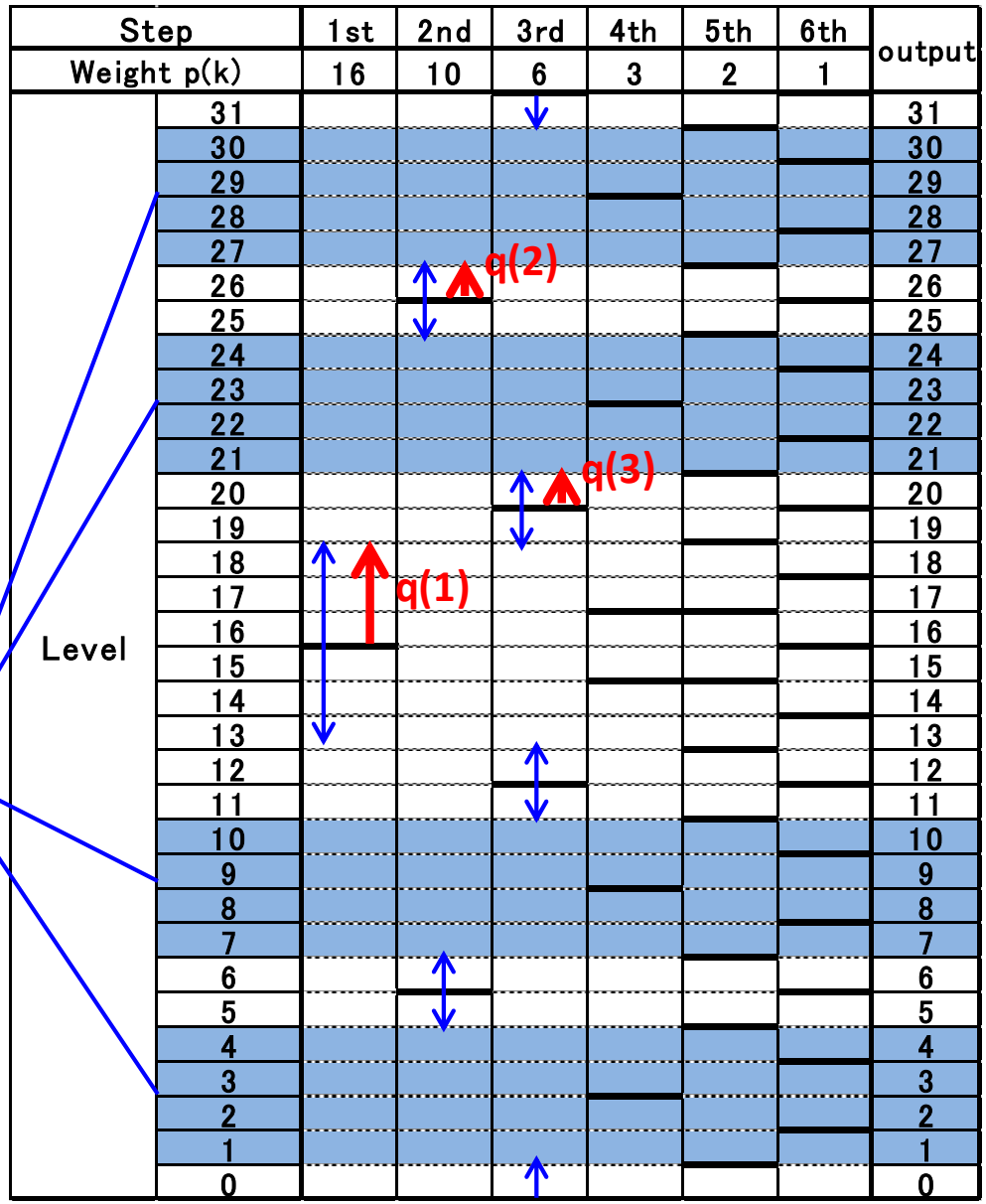
Reference voltage weight p(k)

$$p(1) = 2^{5-1} = 16$$
$$p(2) = 1.8^{4} \cong 10$$
$$p(3) = 1.8^{3} \cong 6$$
$$p(4) = 1.8^{2} \cong 3$$
$$p(5) = 1.8^{1} \cong 2$$
$$p(6) = 1.8^{0} = 1$$

uncorrectable range

Not effective redundancy design

Good selection method is needed !



Outline

- Research Background
- SAR ADC Redundancy Design
- **Proposed SAR Algorithm Using Fibonacci Sequence**
- Advantages of Proposed SAR Algorithm
- Conclusions

Fibonacci Sequence

Definition (n=0,1,2,3…)

$$F_0 = 0$$
$$F_1 = 1$$
$$F_{n+2} = F_n + F_{n+1}$$

Example of numbers(Fibonacci number)

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89…

Leonardo Fibonacci (around 1170-1250)

Property

The closest terms ratio converges to *"Golden Ratio"* !

$$\lim_{n\to\infty} \frac{F_n}{F_{n-1}} = 1.618033988749895 = \varphi$$

Proposed Method of p(k) Selection

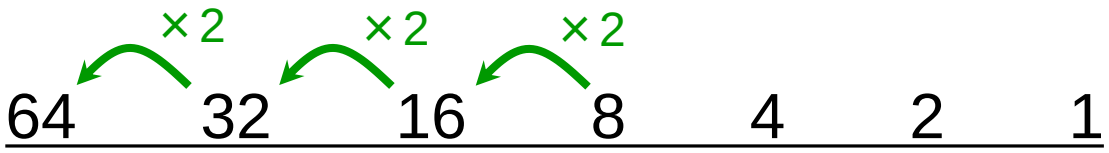
We select N bit and M step SAR ADC k-th step reference voltage p(k).

here $p(1) = 2^{N-1}$

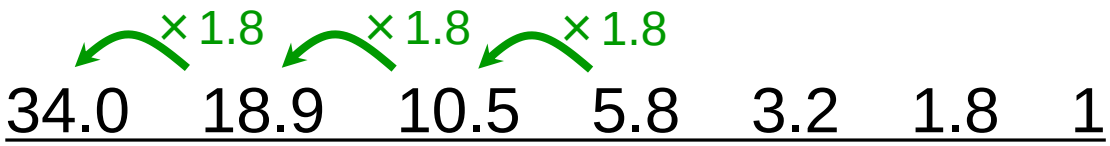
Proposed solution

Using Fibonacci sequence for p(k):$p(k) = F_{M-k+1}$

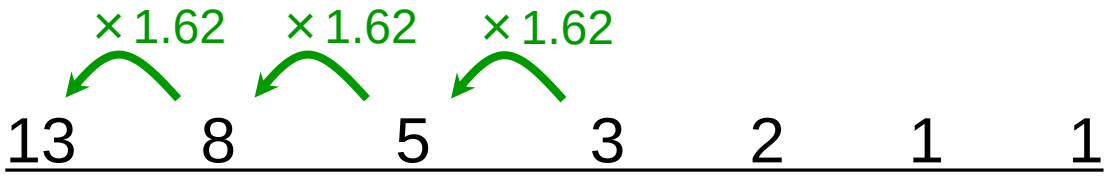
Binary Weight



Radix1.8 Weight



Fibonacci Weight
(Radix 1.62 Weight)



Property converging to Golden Ratio

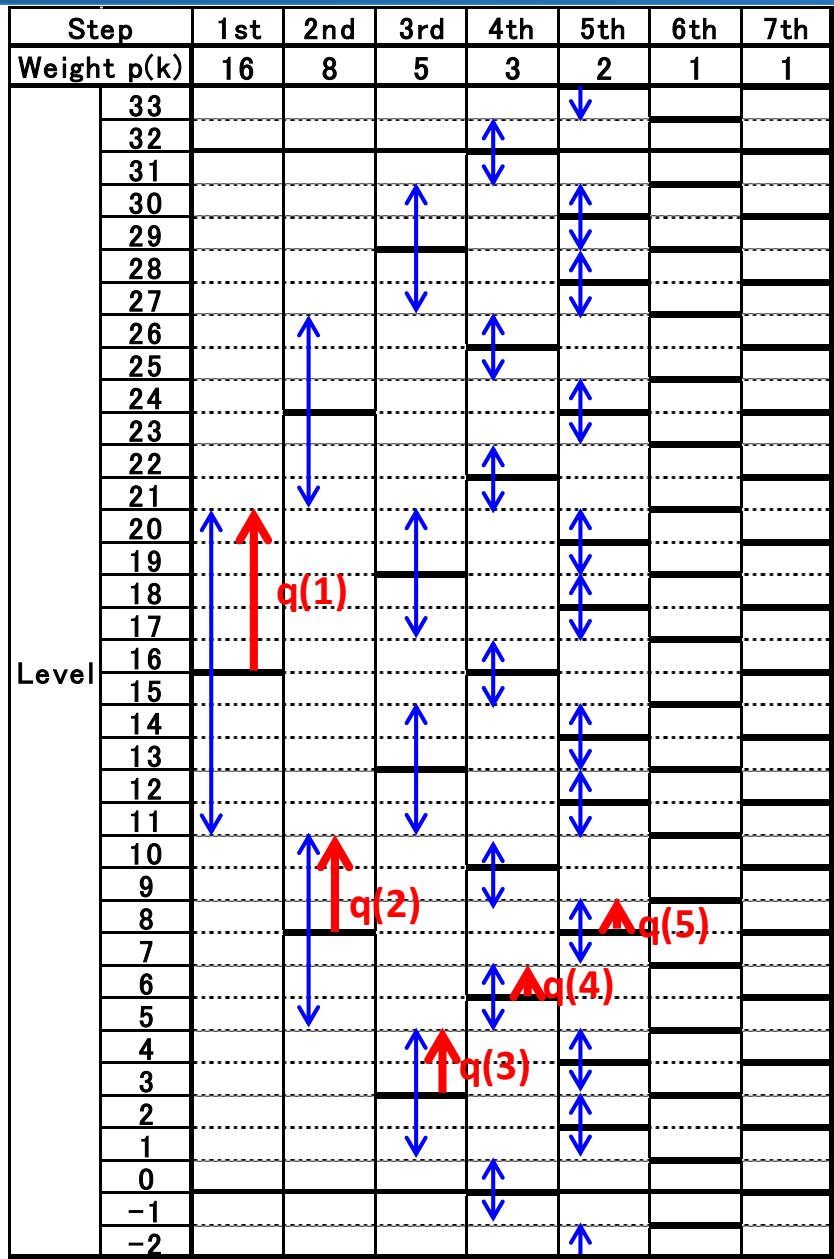
➡ Realize **Radix 1.62 Weight** by using **only integer** !

SAR ADC Using Fibonacci Sequence

Fibonacci sequence SAR ADC

Two properties are discovered !!

1. ***Correctable difference q(k) is always Fibonacci number*** F_{M-k-1}.
2. ***q(k) is exactly in contact q(k+1) without overlap.***



Properties of Fibonacci Sequence Usage

Fibonacci sequence SAR ADC

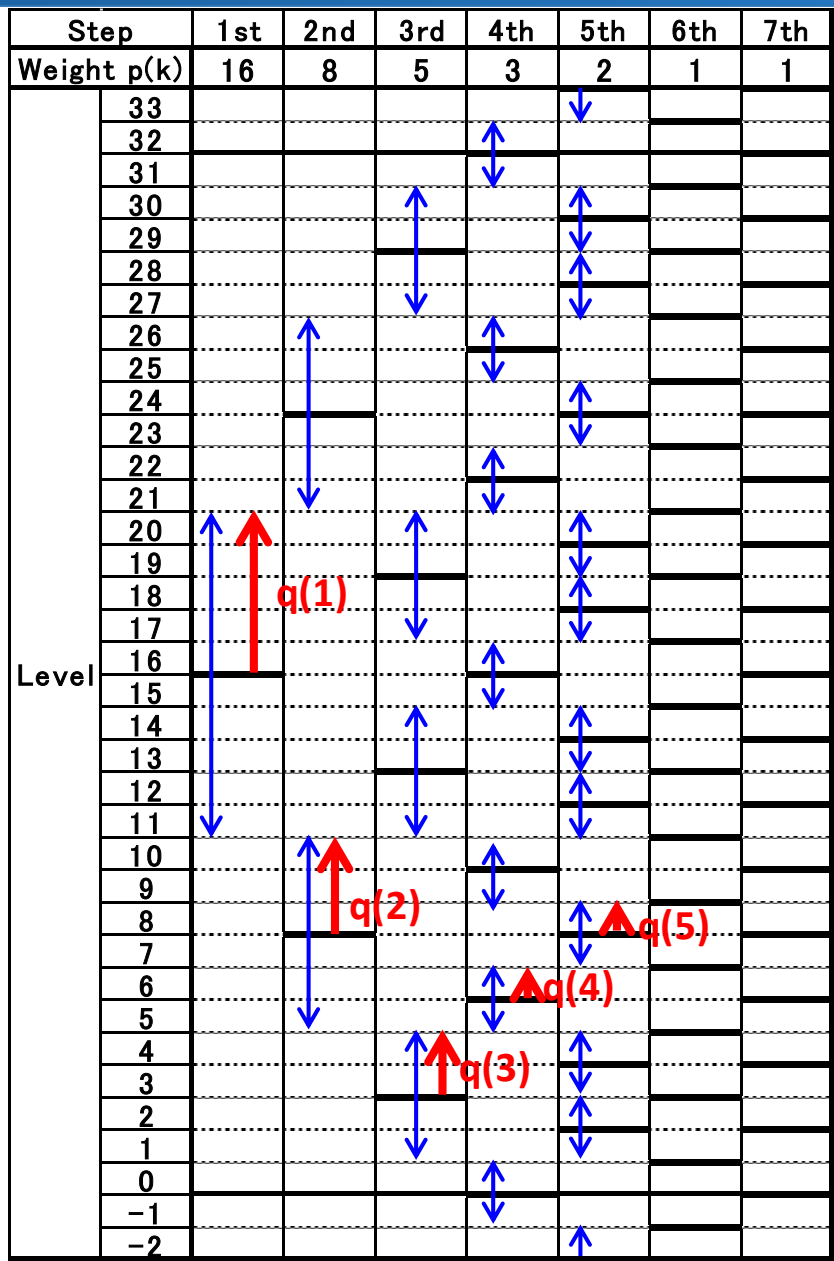
Two properties are discovered !!

1. ***Correctable difference q(k) is always Fibonacci number*** F_{M-k-1}.
2. *q(k) is exactly in contact q(k+1) without overlap.*

Correctable difference q(k)

$$q(1) = 5 = F_5$$
$$q(2) = 3 = F_4$$
$$q(3) = 2 = F_3$$
$$q(4) = 1 = F_2$$
$$q(5) = 1 = F_1$$

For proof, see our paper.



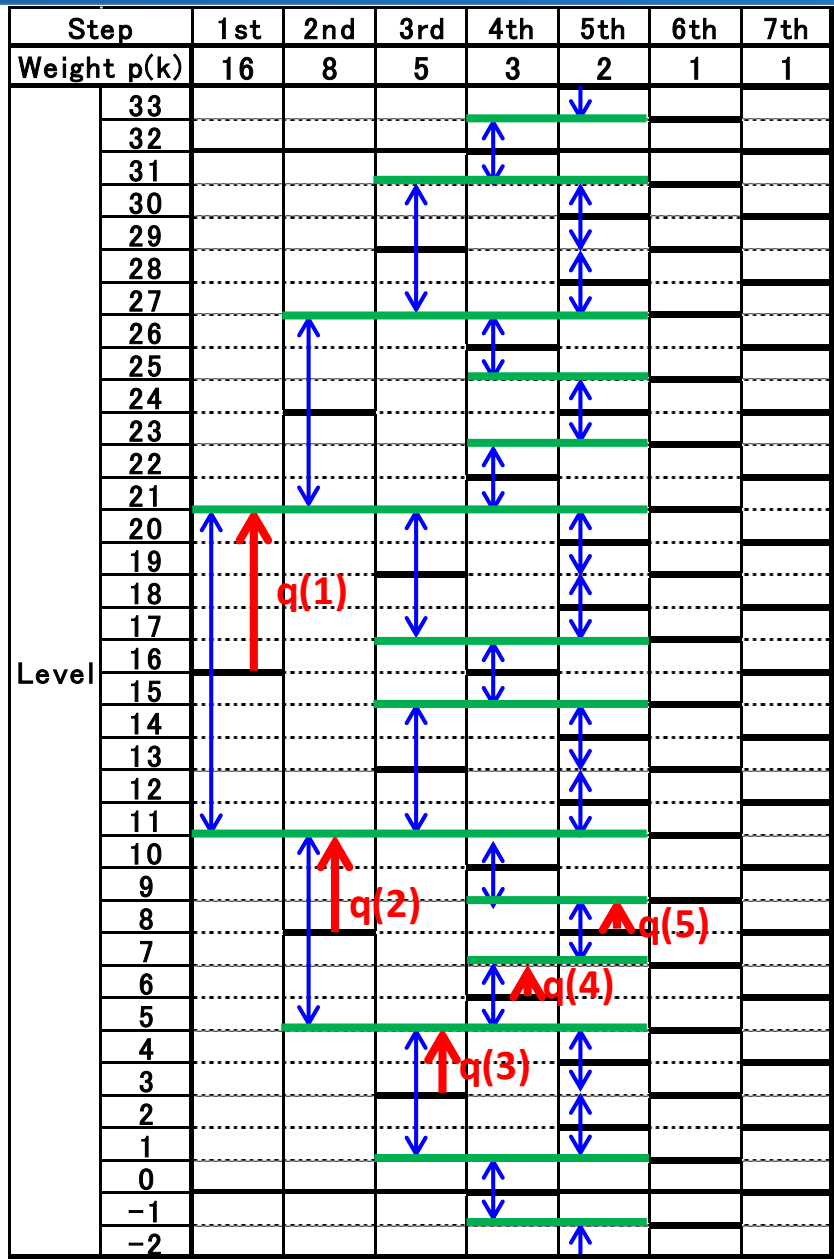
Properties of Fibonacci Sequence Usage

Fibonacci sequence SAR ADC

Two properties are discovered !!

1. *Correctable difference q(k) is always Fibonacci number* F_{M-k-1}.
2. ***q(k) is exactly in contact q(k+1) without overlap.***

For proof, see our paper.



Significance of Property 2

Fibonacci sequence SAR ADC

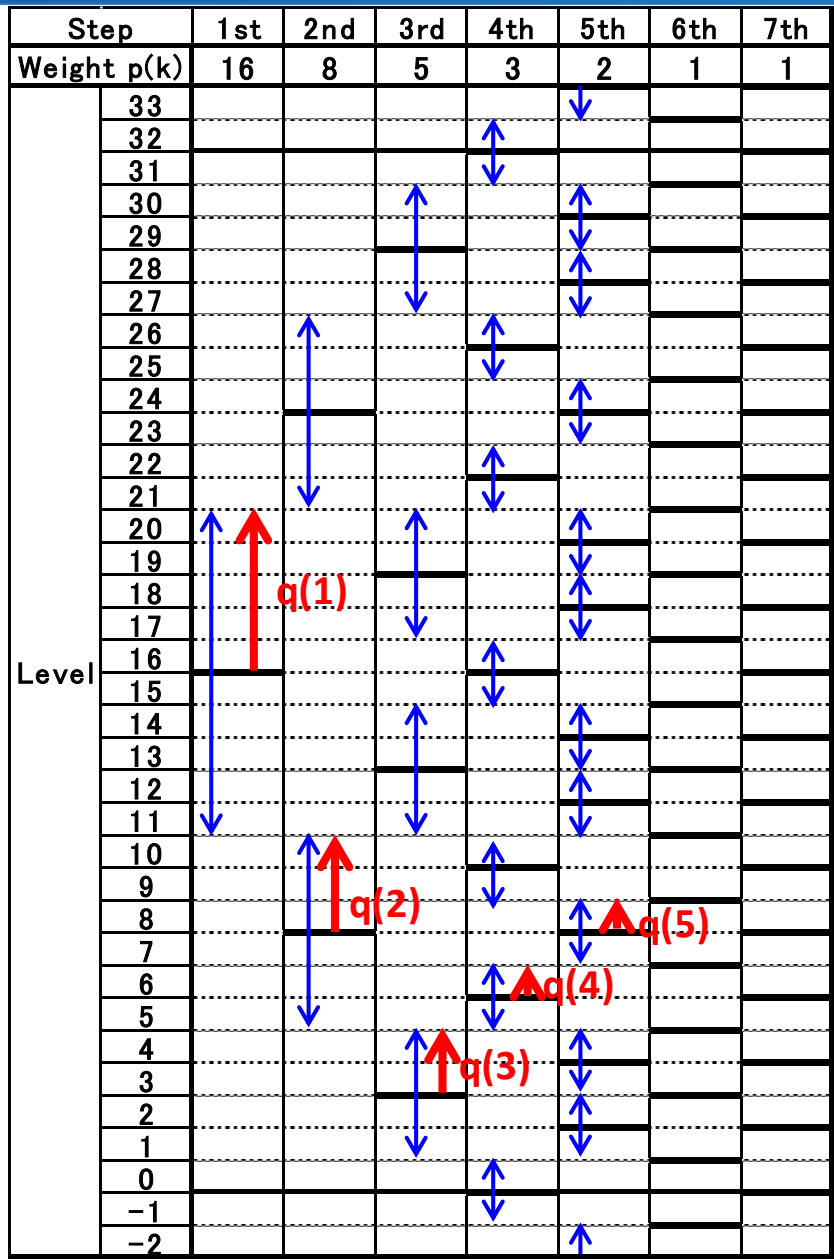
Two properties are discovered !!

1. *Correctable difference q(k) is always Fibonacci number* F_{M-k-1}.
2. ***q(k) is exactly in contact q(k+1) without overlap.***

If radix is **bigger** than 1.62
→ **separated** q(k)

If radix is **smaller** than 1.62
→ **overlapped** q(k)

Golden ratio can establish a standard !



Significance of Property 2

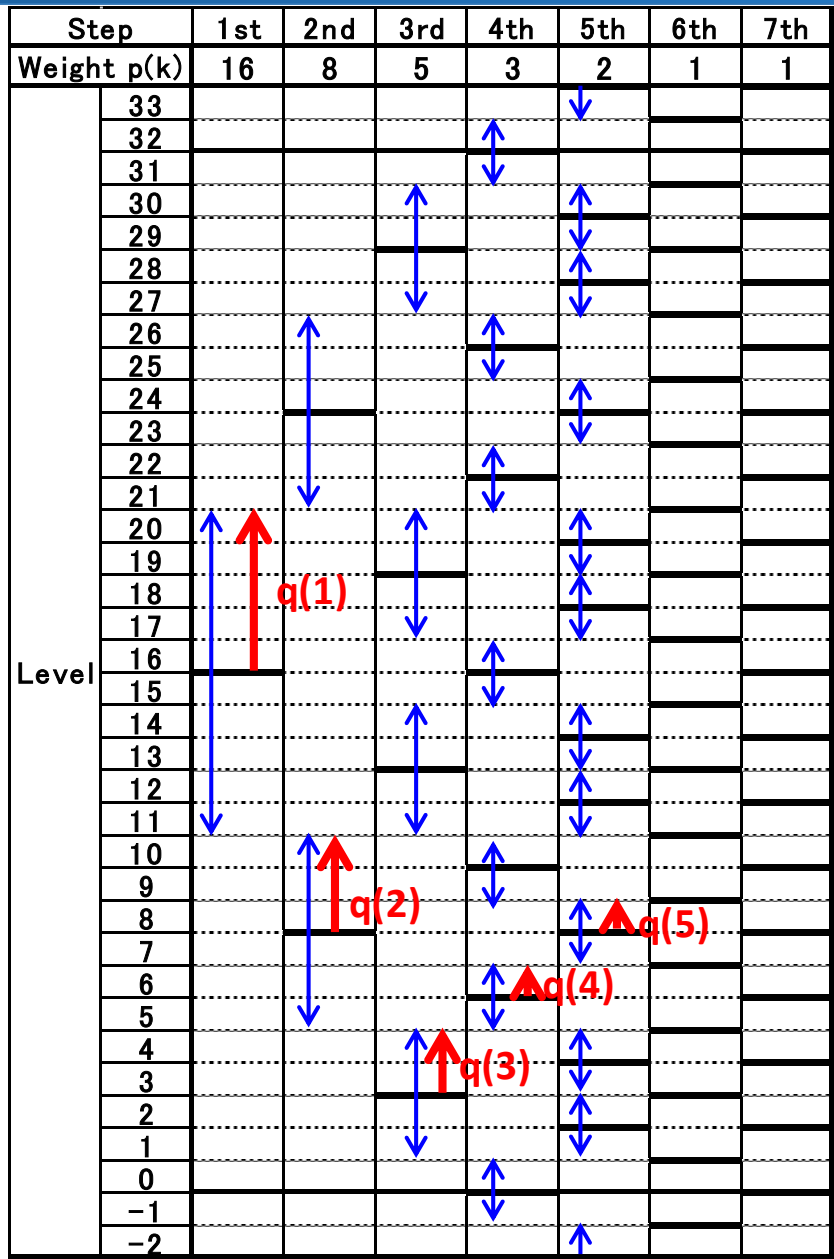
Fibonacci sequence SAR ADC

Two properties are discovered !!

1. *Correctable difference q(k) is always Fibonacci number* F_{M-k-1}.
2. ***q(k) is exactly in contact q(k+1) without overlap.***

Golden ratio covers wide input range by minimum extra comparison steps.

The most efficient design !

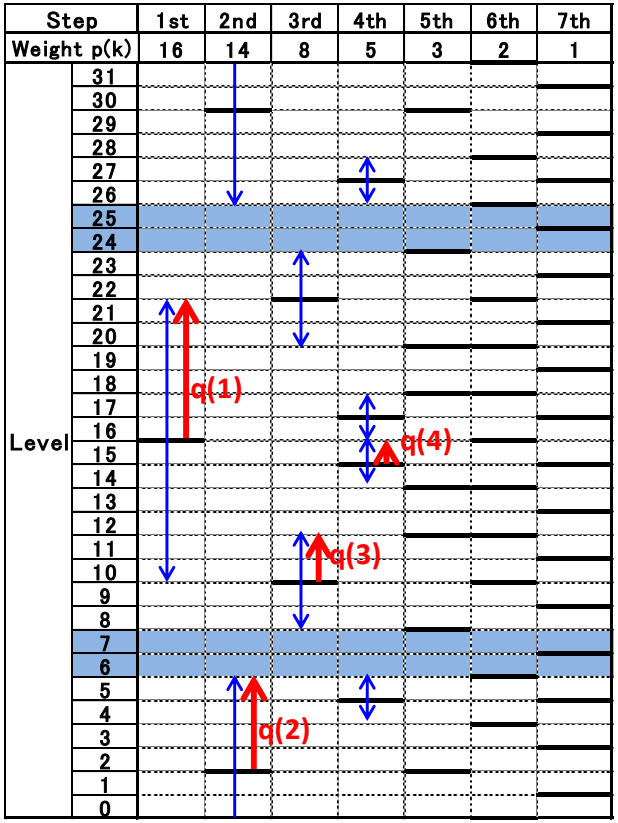


Comparison with Conventional Method

5bit SAR ADC

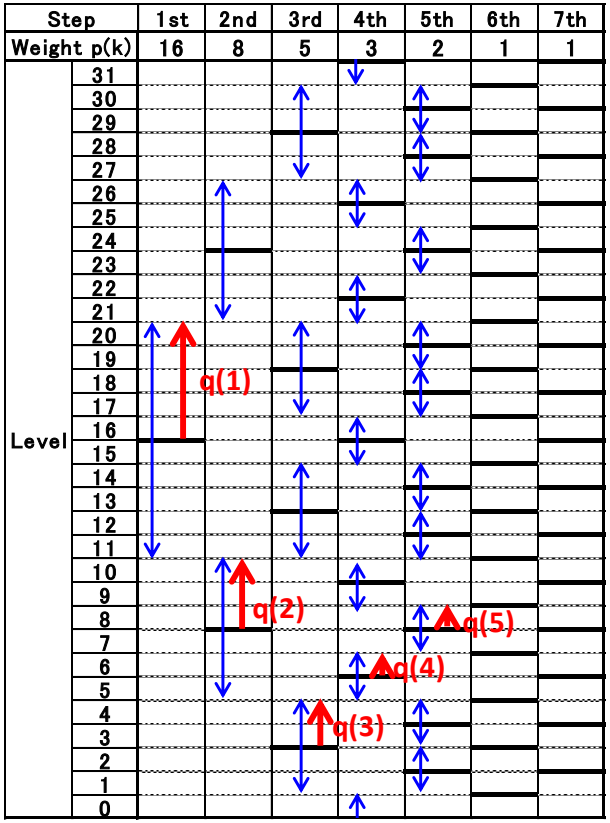
Conventional method

Radix=1.7



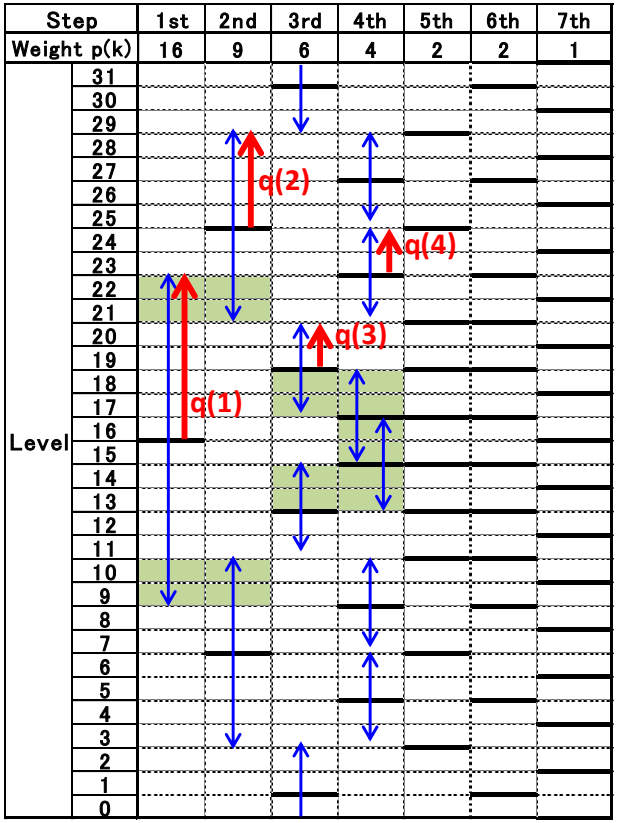
Proposed method

Radix=1.62



Conventional method

Radix=1.55



Redundant design using Fibonacci sequence

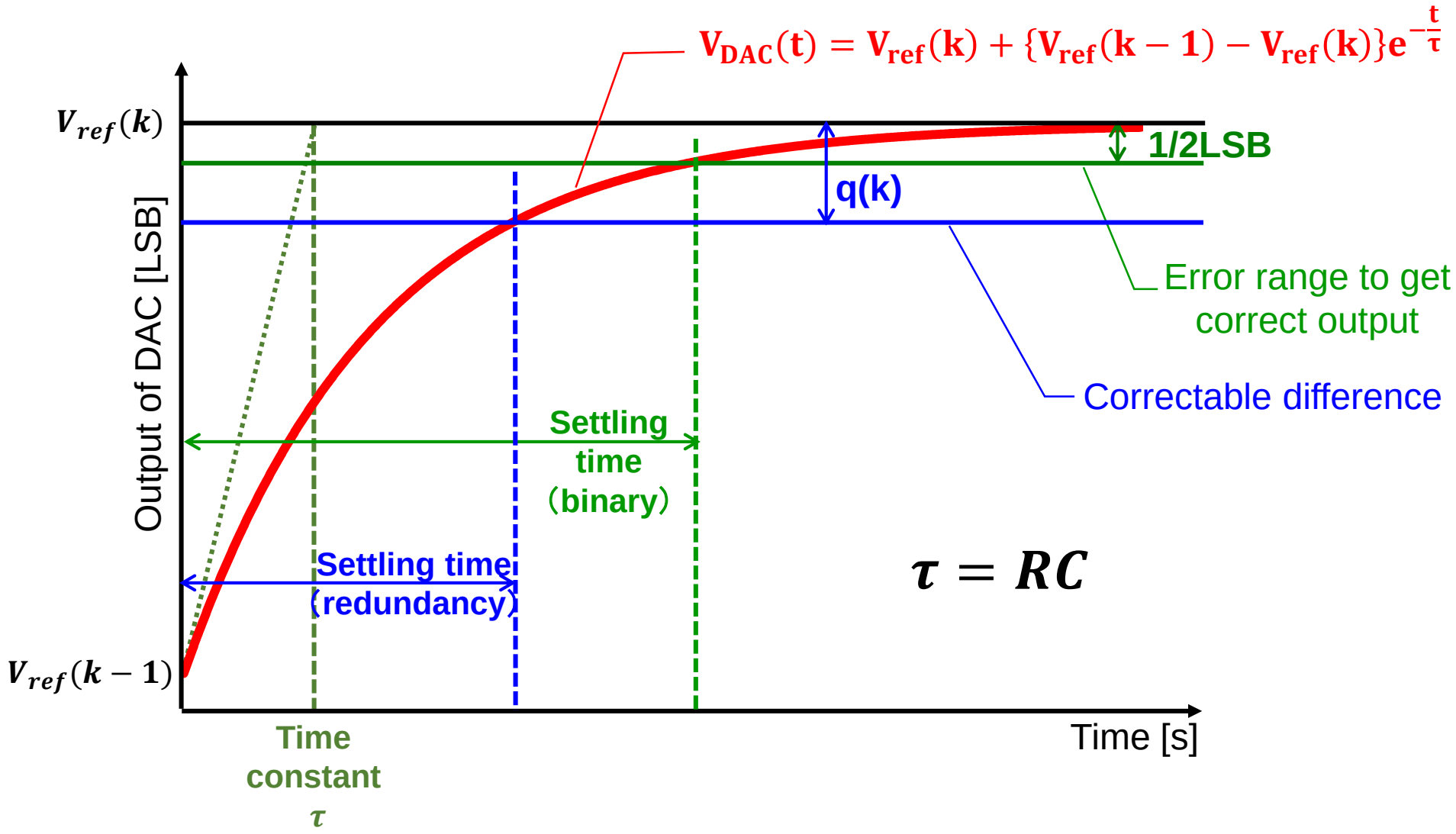
{ Radix standard
Efficiency design

Outline

- Research Background
- SAR ADC Redundancy Design
- Proposed SAR Algorithm Using Fibonacci Sequence
- **Advantages of Proposed SAR Algorithm**
- Conclusions

DAC Settling Time

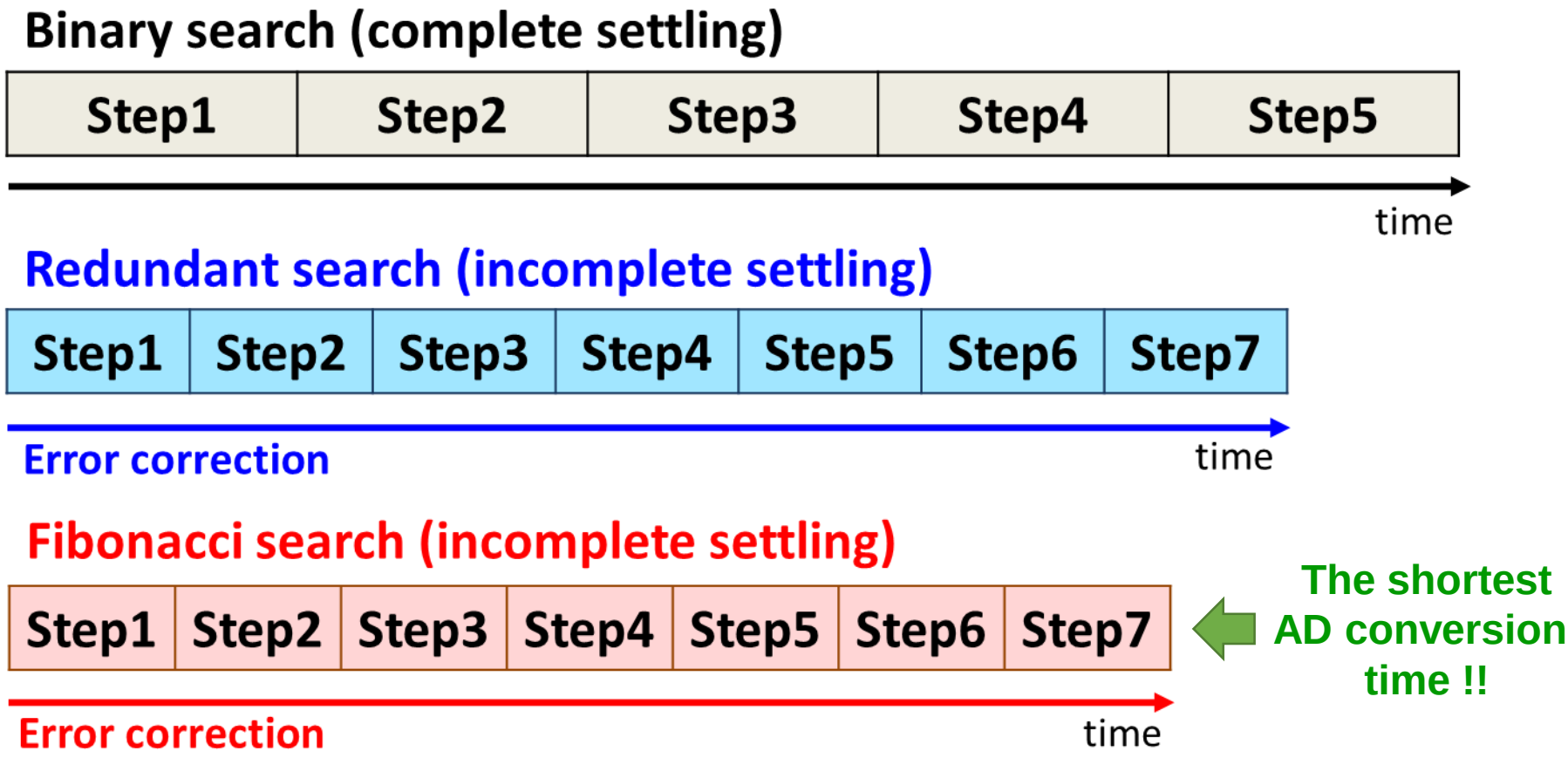
DAC Settling model by using a simple RC model



Reduction of Settling Time

Digital error correction ⇒ **Incomplete settling**

5bit SAR ADC

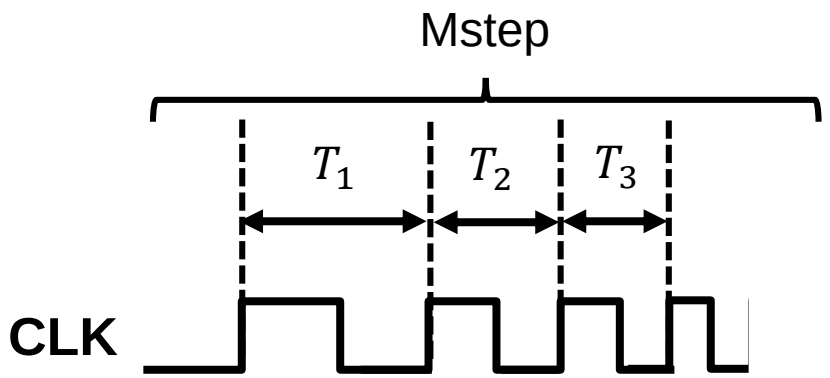


Generalization of DAC Incomplete Settling

Settling time

$$T_{settle}(k) = \tau \ln\left(\frac{p(k) + q(k-1)}{q(k)}\right)$$

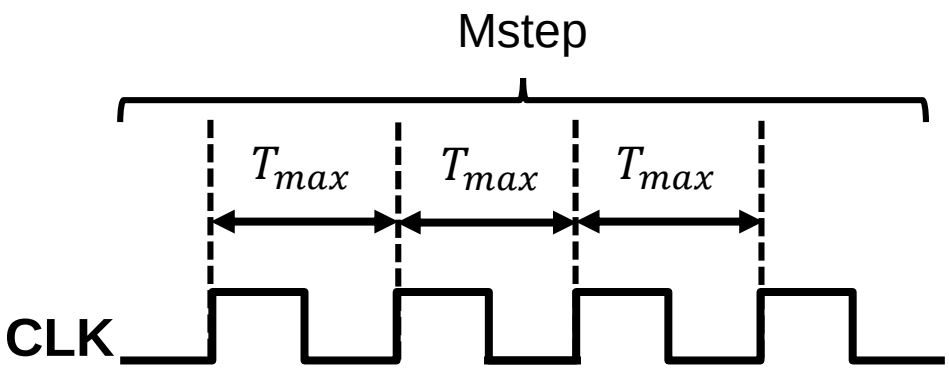
Variable clock SAR ADC



Total settling time

$$T_{total\_vari} = \sum_{i=1}^{M} T_i$$

Fixed clock SAR ADC

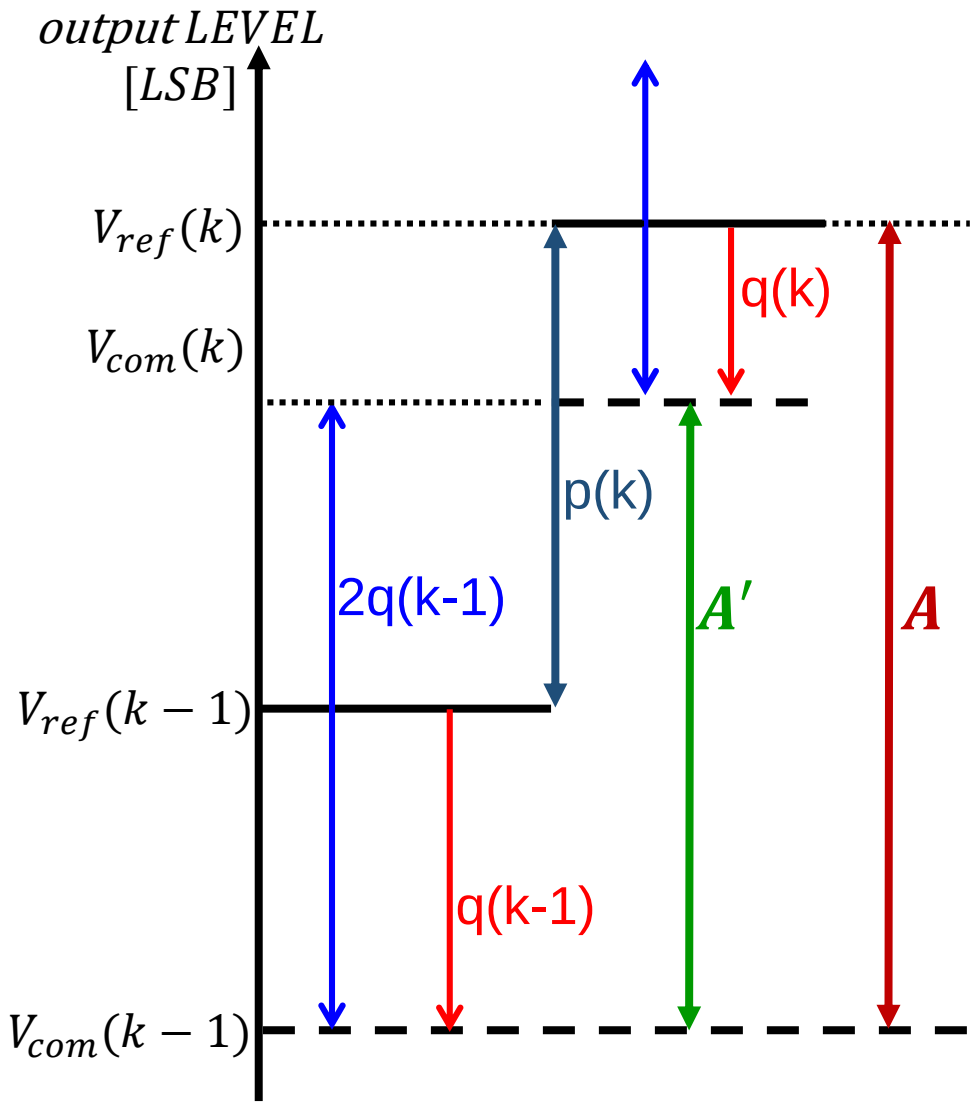


Total settling time

$$T_{total\_fixed} = T_{max} \times M$$

Settling Time in Fibonacci Case

Fibonacci sequence settling time



$$T_{settle}(k) = \tau \ln\left(\frac{p(k)+q(k-1)}{q(k)}\right)$$

using Fibonacci sequence

$$p(k) = F_{M-k+1}$$
$$q(k) = F_{M-k-1}$$

$$= \tau \ln\left(\frac{F_{M-k+1}+F_{M-k}}{F_{M-k-1}}\right)$$

$$= \tau \ln\left(\frac{(F_{M-k}+F_{M-k-1})+F_{M-k}}{F_{M-k-1}}\right)$$

$$= \tau \ln\left(2\frac{F_{M-k}}{F_{M-k-1}}+1\right)$$

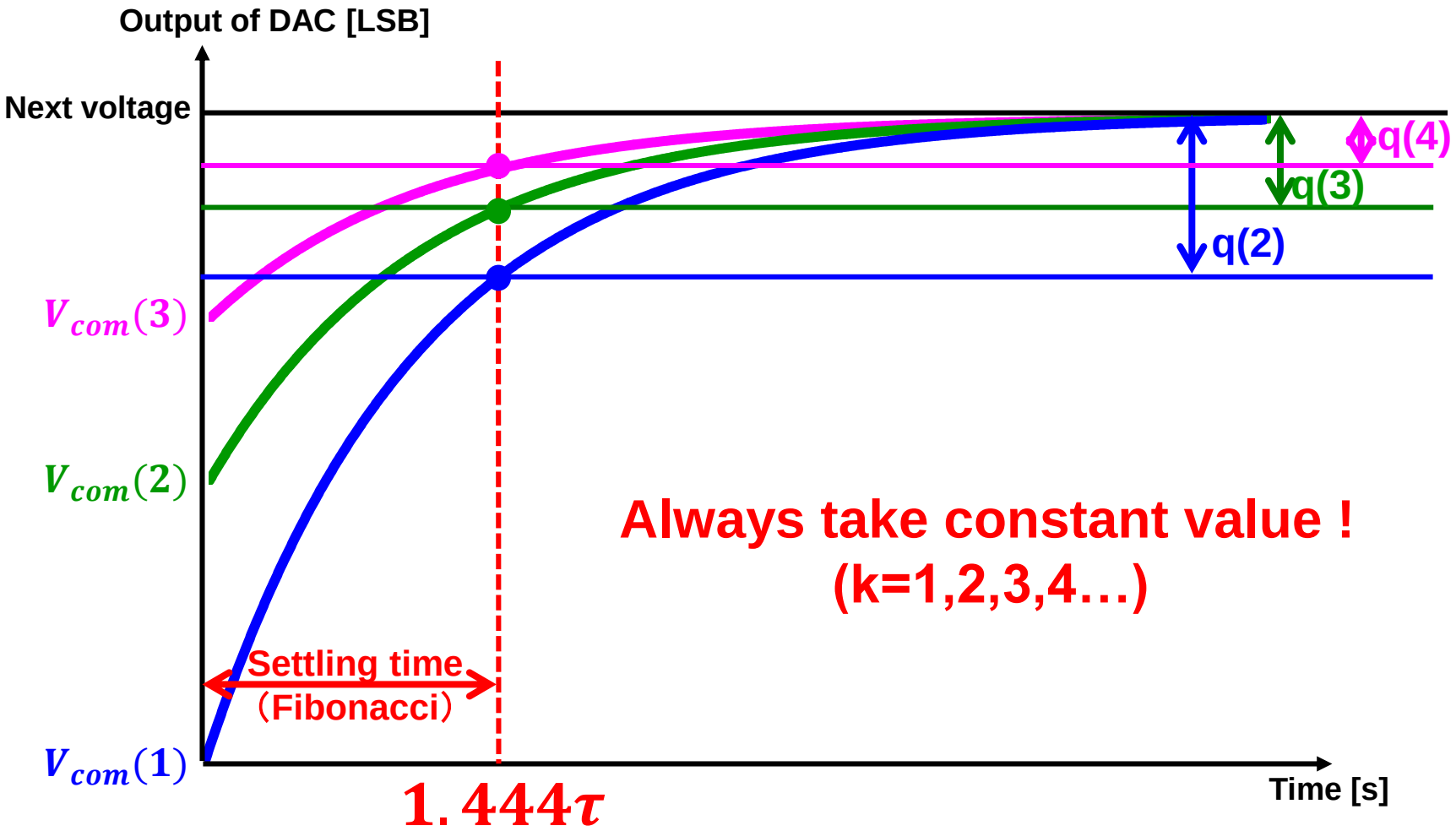
$$\lim_{n\to\infty}\frac{F_n}{F_{n-1}} = 1.618 = \varphi$$

$$T_{settle}(k) = \tau \ln(2\varphi+1) = 1.444\tau$$

Settling Time in Fibonacci Case

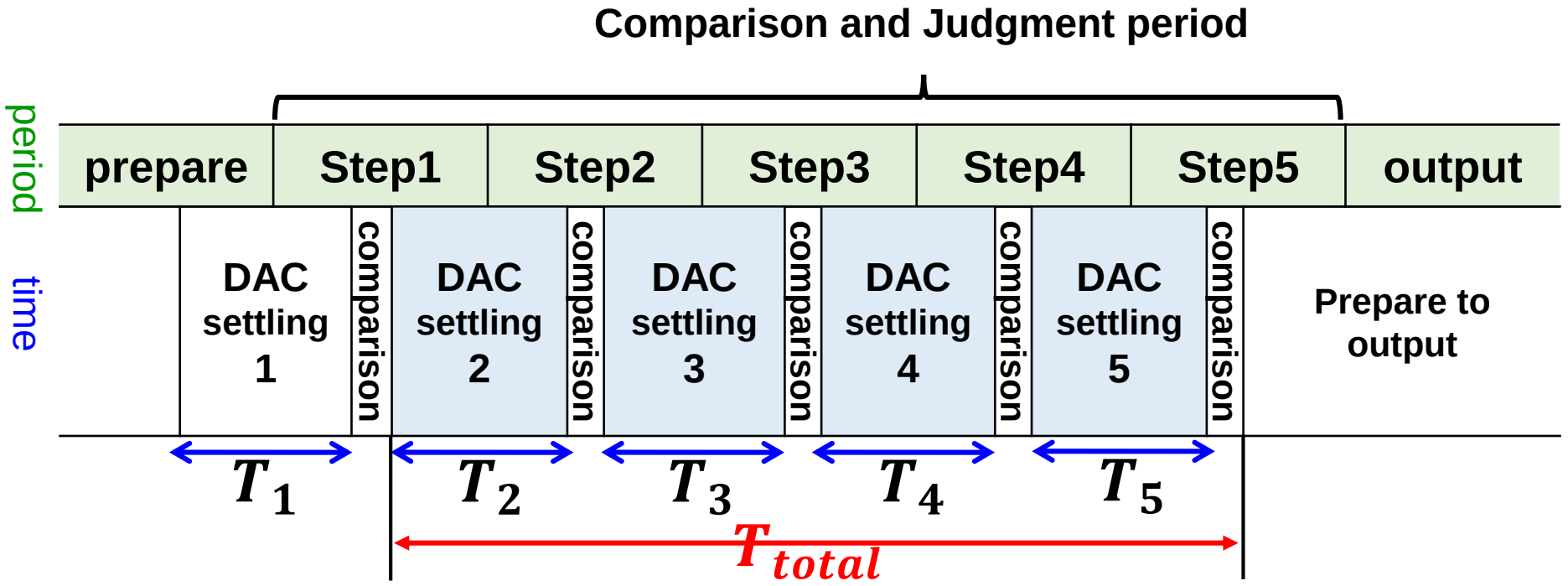
New property is discovered !

$$T_{settle}(k) = \tau \ln(2\varphi + 1) = 1.444\tau \quad \text{for all k}$$



Simulation of Total Settling Time

5step SAR ADC

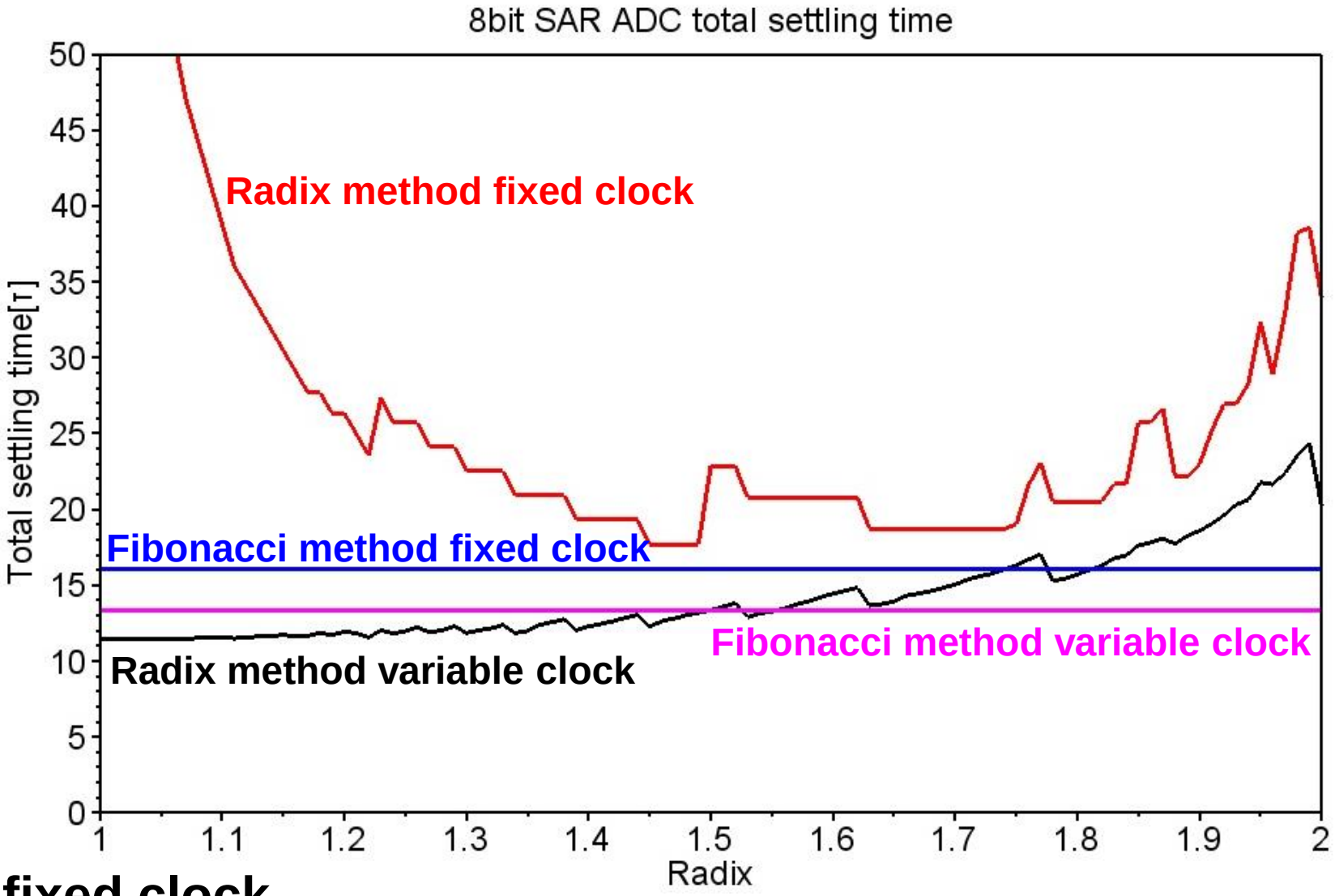


Conditions of simulation

- Investigate total settling time T_{total} of Variable clock and Fixed clock

$$T_{settle}(k) = \tau \ln\left(\frac{p(k)+q(k-1)}{q(k)}\right), \qquad T_{settle}(2) = \tau \ln\left(\frac{p(k)}{q(k)}\right)$$

Result of Fibonacci SAR ADC Settling

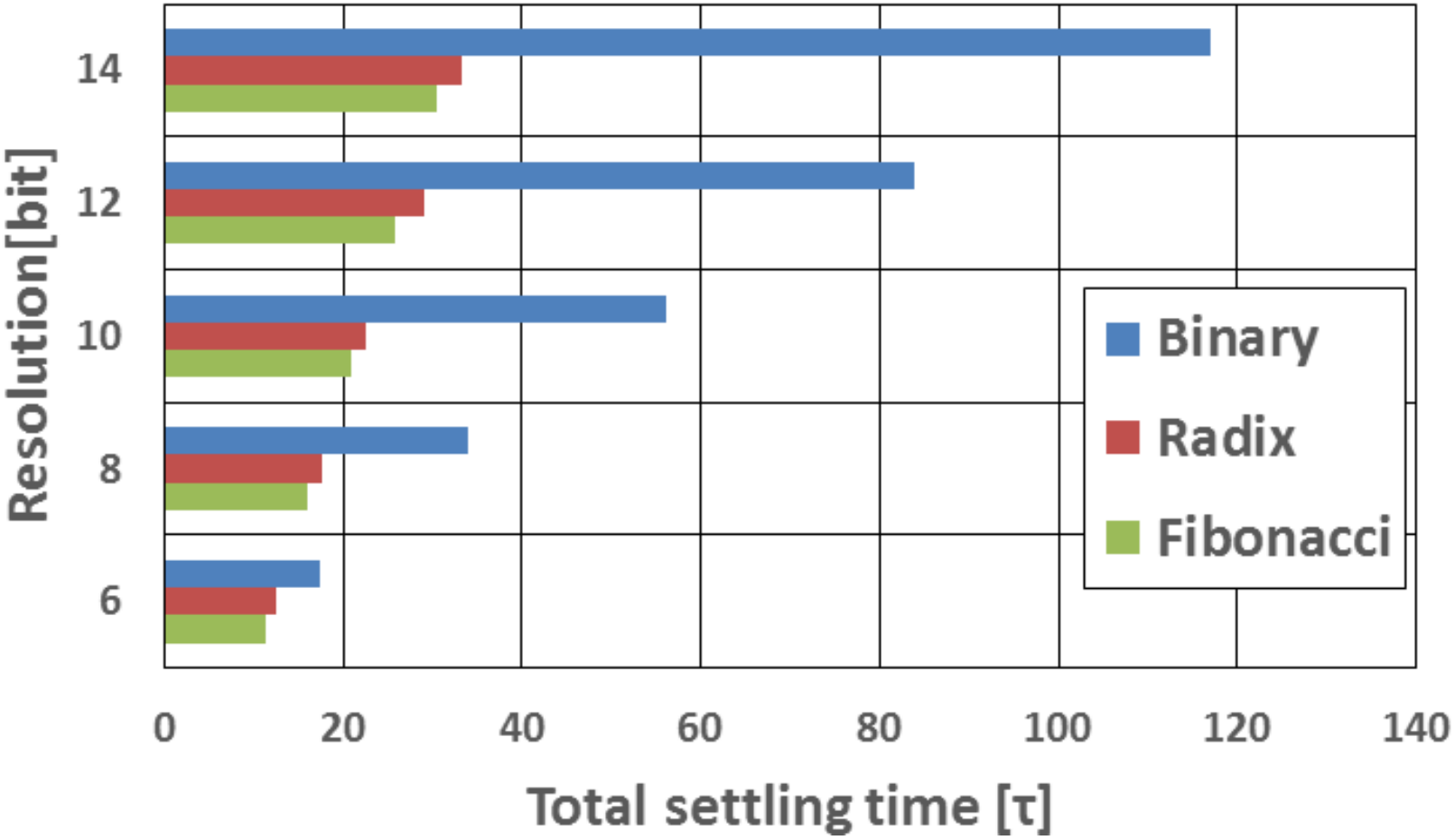


At fixed clock,

Fibonacci SAR ADC realizes the shortest settling time !!

Summary of Result

Result of each resolution at fixed clock



At fixed clock,

Fibonacci SAR ADC is faster than Radix SAR ADC !

Conclusion

- Propose redundant SAR ADC design methods
- Get important properties by using Fibonacci sequence
  - **Reliable**
    Correctable difference covers wide input range
  - **Shortest-Conversion**
    Conversion time is shortest in a fixed clock
  - **Radix-Standard**
    Golden ratio φ establish radix standard
  - **Constant-Settling**
    DAC settling time is constant for each step

Number Theory for Engineering

Carolus Fridericus Gauss
(1777-1855)

"Number theory is the queen of mathematics"

Carolus Fridericus Gauss

Past Number theory

Beautiful and Mysterious
was NEVER practical

Current Number theory

used information communication processing

➡ good match to Digital technology

Number theory application for ADC/DAC is a frontier.
There are great chances for new discovering !

質疑応答

堀田先生(東京都市大学)

Q.実現するのに最も難しい部分はどこか？

A.DACの正確性が最も重要です。それは冗長設計SARADCはDACのミスマッチを補正することができないからです。

Q.バリアブルクロックはどのように作成するか？

A.今のところ考えていません。今後の実装の問題になると考えています。

Truong Quang Vinhさん(ベトナムナショナルユニバーシティ)

Q.ゲートの数はどのくらいになりますか？回路面積は何パーセントくらい増えそうですか？

A.データがなく実際に作らないとわかりません。しかしほとんどをディジタル回路で作成しているので、そこまで大きな余分とはならないはずです。

Appendix

Lucas sequence

Definition (n=0,1,2,3…)

$$L_0 = 2$$
$$L_1 = 1$$
$$L_{n+2} = L_n + L_{n+1}$$

Example of numbers(Lucas number)

2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123…

Édouard Lucas (1842-1891)

Property

The closest terms ratio converges to *"Golden Ratio"* !

$$\lim_{n\to\infty} \frac{L_n}{L_{n-1}} = 1.618033988749895 = \varphi$$

Can we use Lucas sequence?

$$p(k) = L_{M-k}$$

$(here\ p(1) = 2^{N-1}, p(M-1) = L_0, p(M) = L_1)$

SAR ADC using Lucas sequence

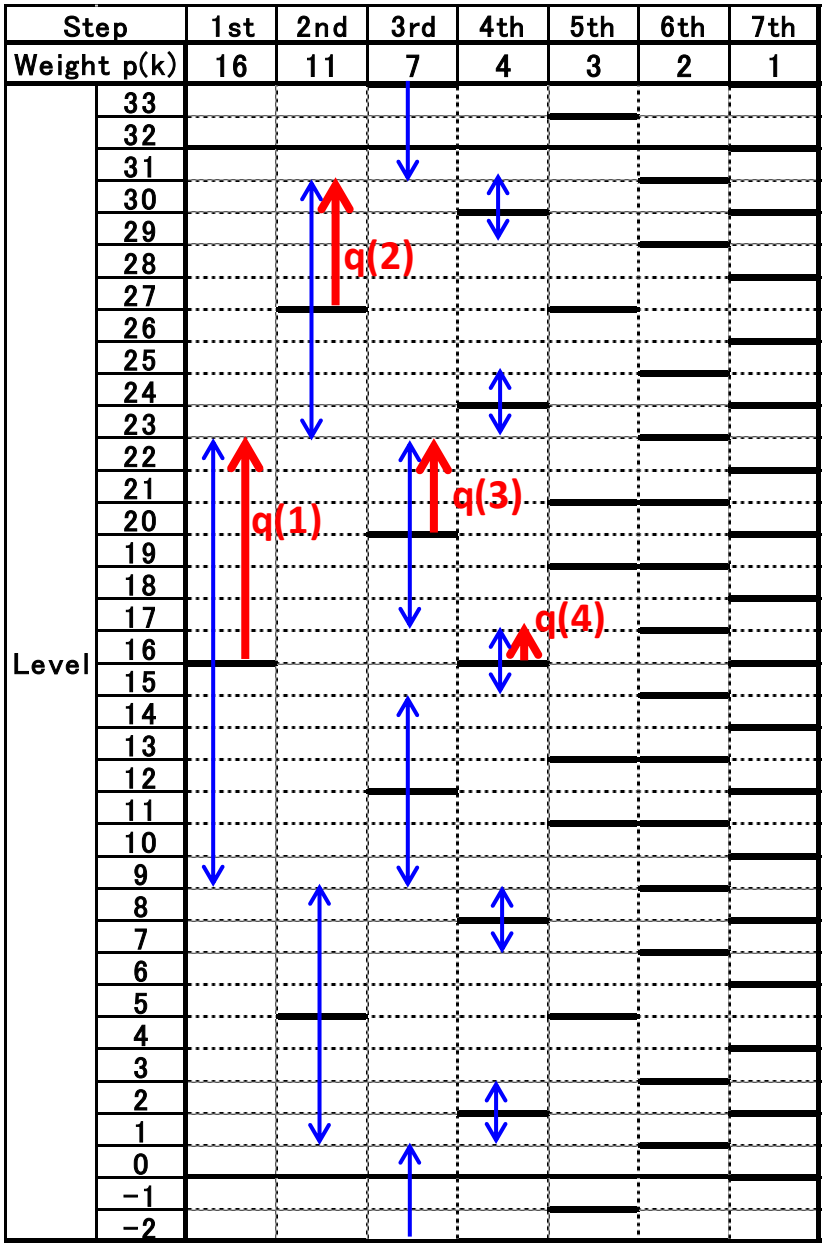
Lucas sequence SAR ADC

Two properties are discovered !!

1. ***Correctable difference q(k) is always Lucas number L_{M-k-2}.***
2. ***q(k) is exactly in contact q(k+1) without overlap.***

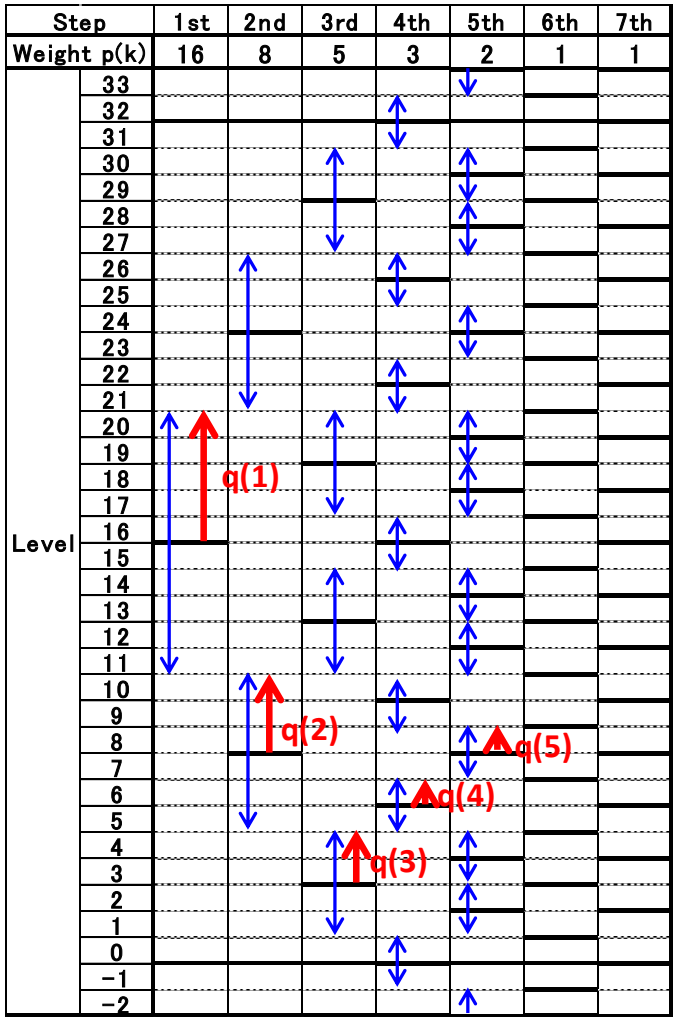
Reconfirming

Radix of Golden ratio is boundary.

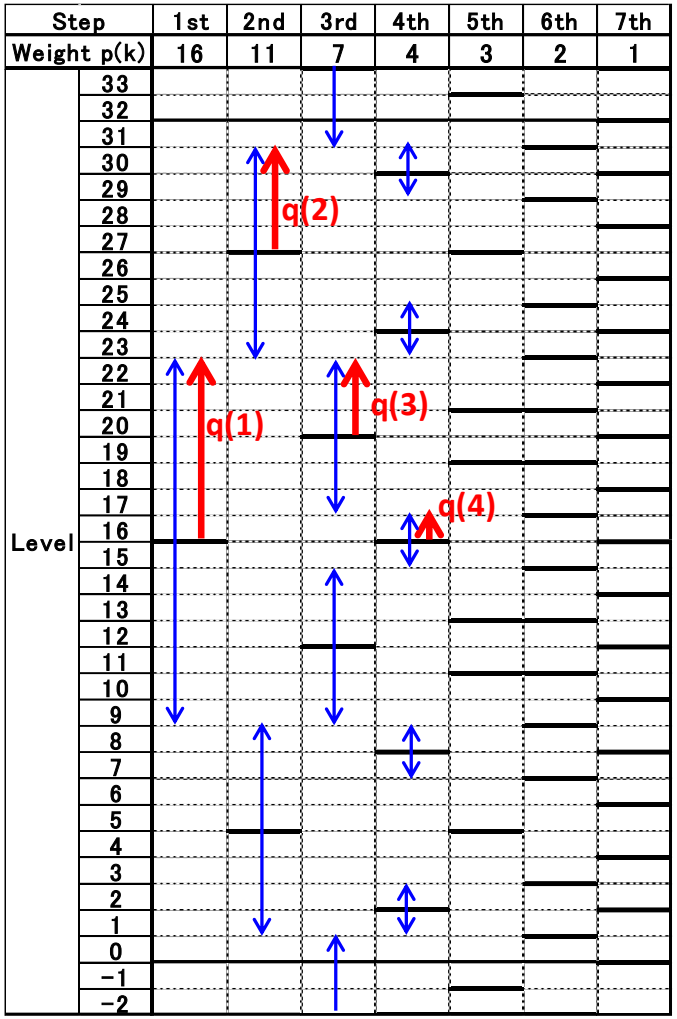


Fibonacci sequence vs. Lucas sequence

Fibonacci SAR ADC

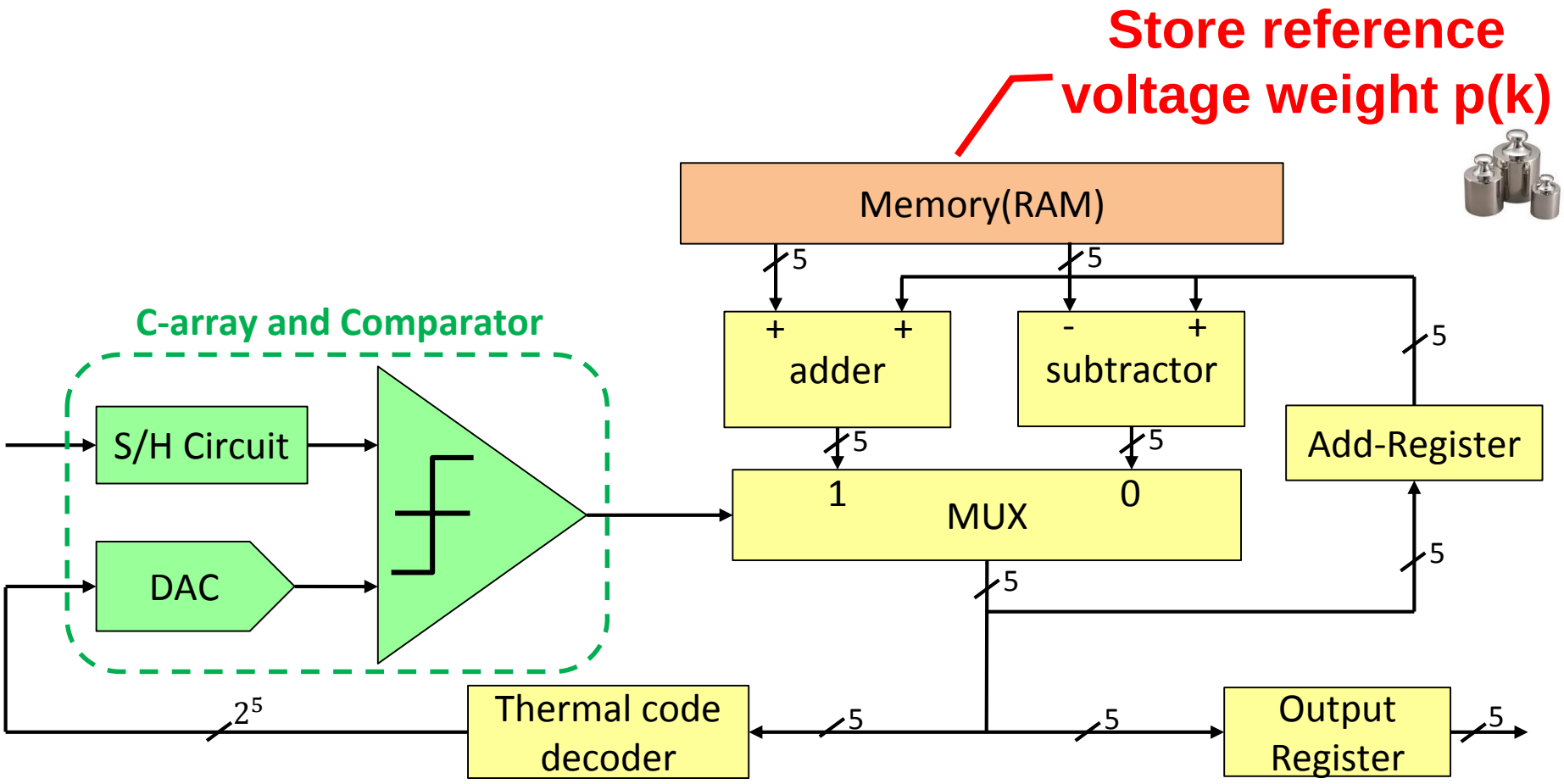


Lucas SAR ADC



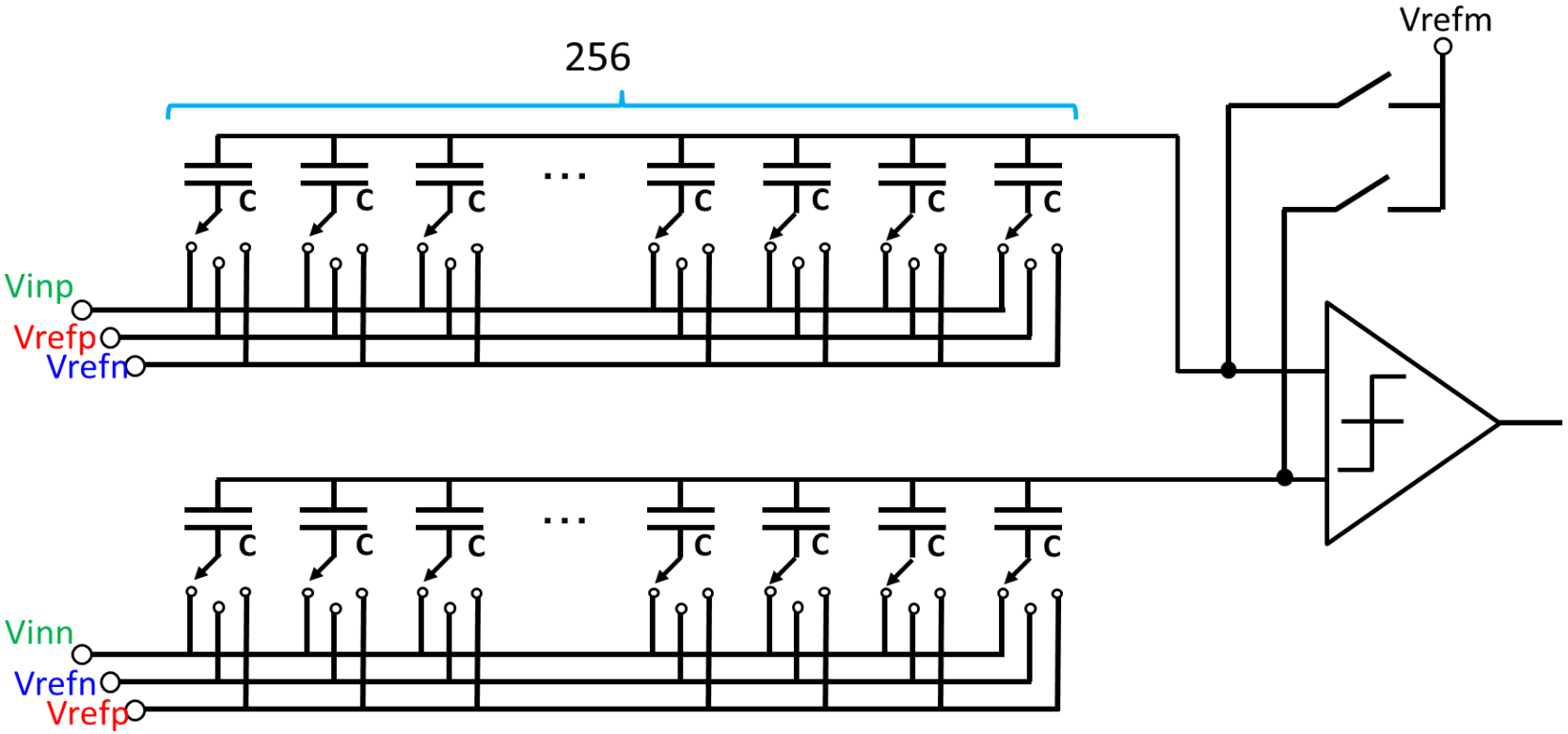
Using Number theory is effective method !

Configuration Block of redundant SARADC



Almost Construction Consist of Digital circuit

C-array and comparator block



upper

Sample Vinp ⇒compare to $\frac{n}{2^N}[LSB]$ n-cap Vrefp , $(2^N - n)$-cap Vrefn

lower

Sample Vinn ⇒compare to $\frac{n}{2^N}[LSB]$ n-cap Vrefn , $(2^N - n)$-cap Vrefp

Generalization of DAC incomplete settling

Consider the longest settling time

Kind of code immigration

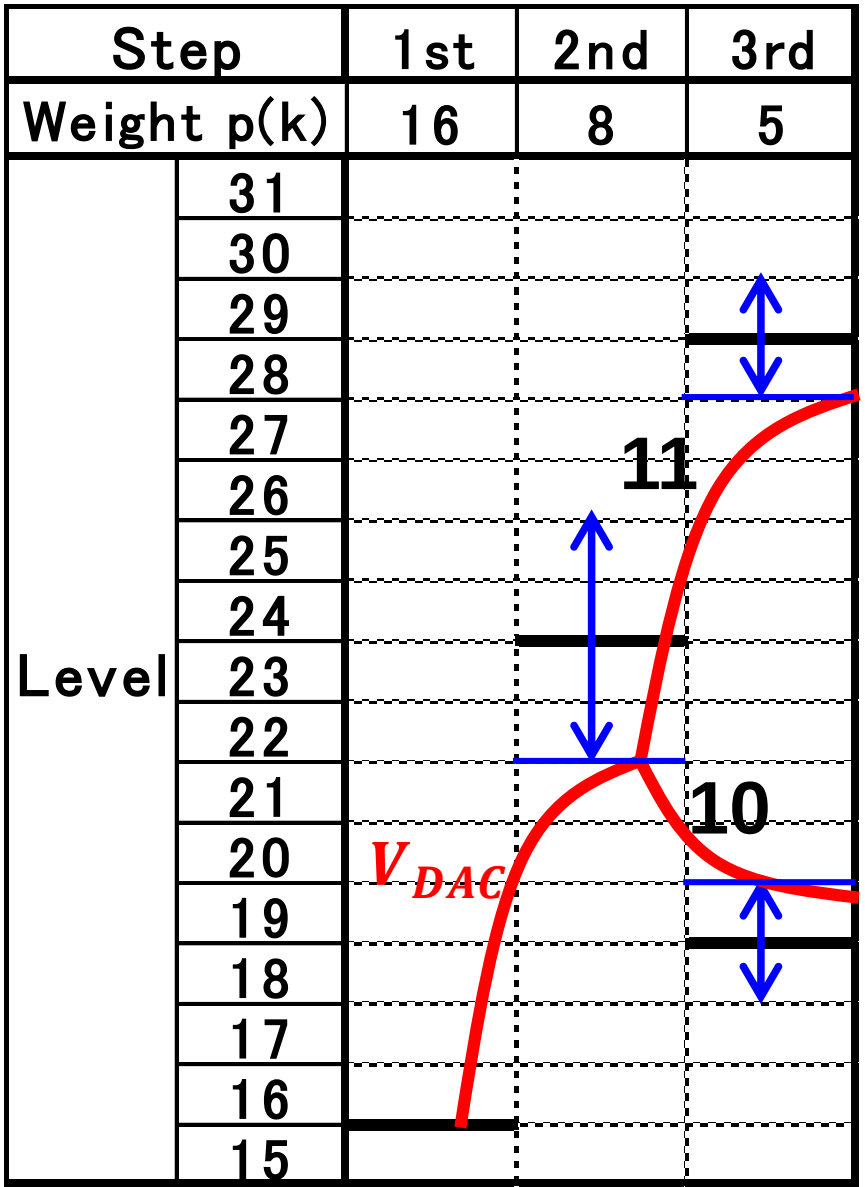
XX or XY

***X,Y are 0 or 1**

Case of XX: longest settling

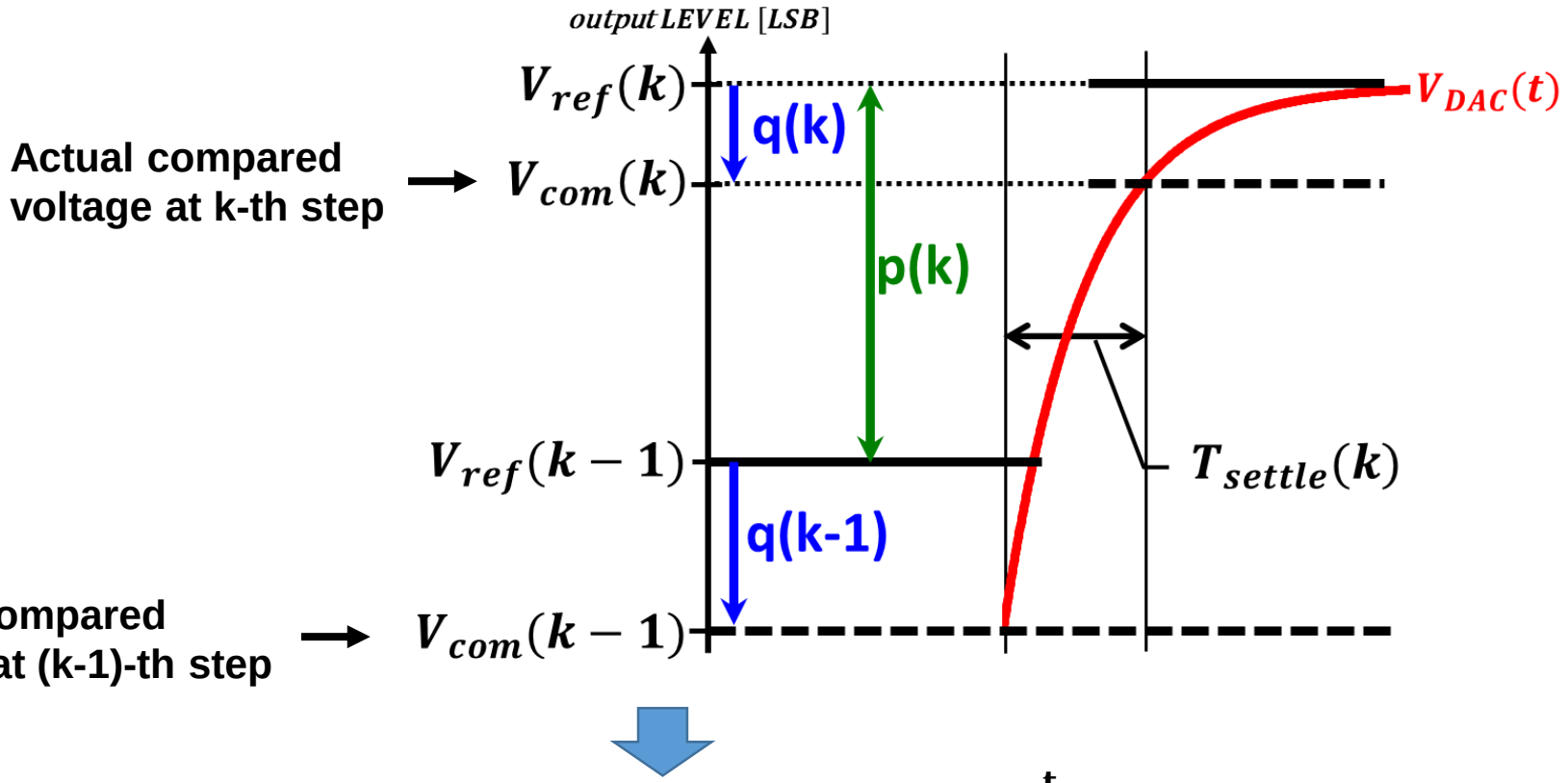
Need XX type settling time for accurate conversion

Radix 1.70 method



Generalization of DAC incomplete settling

Consider the longest settling time



$$V_{DAC}(t) = V_{ref}(k) + \{V_{ref}(k-1) - V_{ref}(k)\}e^{-\frac{t}{\tau}} = V_{com}(k)$$

General expression of settling time

$$T_{settle}(k) = \tau \ln\left(\frac{p(k) + q(k-1)}{q(k)}\right)$$