

# 温故知新：古典数学の掘り起しと AD/DA変換器設計への応用

小林春夫  
群馬大学

Kobayashi k\_haruo@el.gunma-u.ac.jp  
Laboratory



# 発表内容

- はじめに
- フィボナッチ数列と黄金比  
逐次比較近似AD変換器 冗長アルゴリズム
- 魔方陣  
セグメント型DA変換器 レイアウト
- 剰余系(孫子算経)  
時間デジタル変換回路
- まとめ

# 発表内容

- はじめに
- フィボナッチ数列と黄金比  
逐次比較近似AD変換器 冗長アルゴリズム
- 魔方陣  
セグメント型DA変換器 レイアウト
- 剰余系(孫子算経)  
時間デジタル変換回路
- まとめ

# ご紹介する内容

講演者の研究室で行っている  
整数論を用いたAD/DA変換器、時間デジタイザ回路の  
アーキテクチャ・レイアウト技術研究



全く新しいアプローチ



「アナログ回路は技術的に面白い」  
の一側面を示したい

整数論: 数学で 最も簡単に見えて 最も奥が深い



群馬大学 電子情報部門 小林研究室  
数学的アプローチ回路設計グループ

# フィボナッチ数列と黄金比

## 逐次比較近似AD変換器 冗長アルゴリズム

小林佑太郎  
楊志翔

Shaiful Nizam Mohyar  
小林春夫



Kobayashi  
Laboratory



群馬大学  
GUNMA UNIVERSITY

# フィボナッチ数列とは

## フィボナッチ数列

$$F_0 = 0$$

$$F_1 = 1$$

$$F_{n+2} = F_n + F_{n+1}$$

## 初めの項を計算

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...

⇒ フィボナッチ数と呼ばれる



Leonardo Fibonacci  
(伊: 1170-1250年頃)

## 隣り合う項の比率は以下の値 $\varphi$ に収束

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

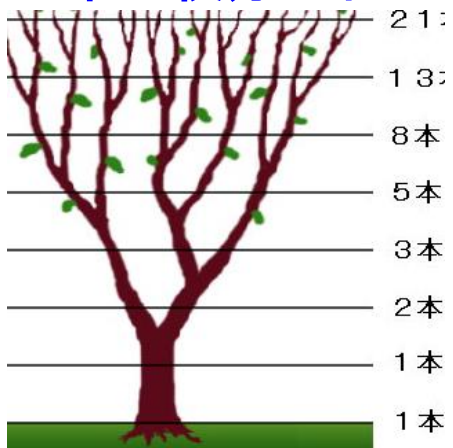
収束比率  $\varphi$  : 黄金比 (Golden ratio)

# 不思議な数“フィボナッチ数”

フィボナッチ数 : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...

植物と関係が深い⇒最も日光を浴びれる、最も多く配置できる

木の枝分かれ



花びら枚数

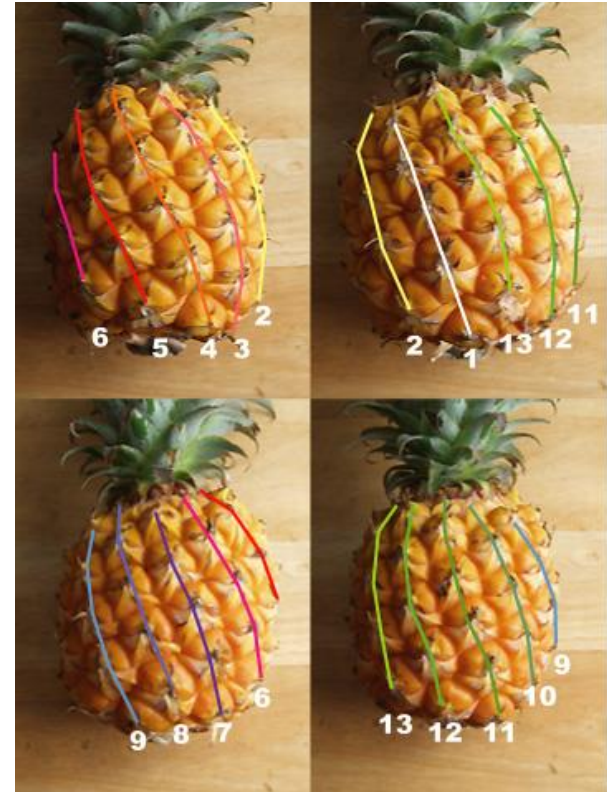


実の付け方

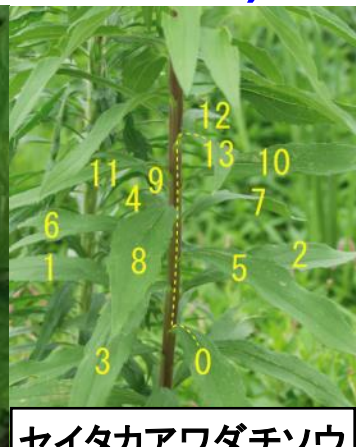
松ぼっくり



パイナップル



葉序(植物の葉の付け方)



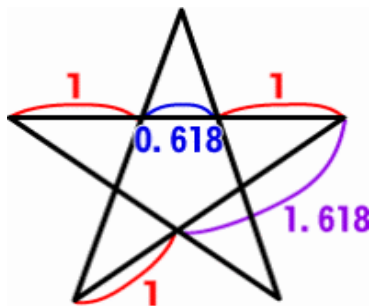


# 最も美しい比率“黄金比”

$$\text{黄金比: } \lim_{n \rightarrow \infty} \frac{F_n}{F_{n-1}} = 1.618033988749895 = \varphi$$

フィボナッチ・黄金比⇒“最も安定し効率の良い配置” 自然の知恵

正五芒星



クフ王のピラミッド(埃)



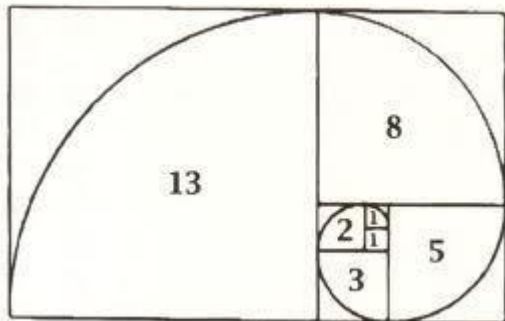
パルテノン神殿(希)



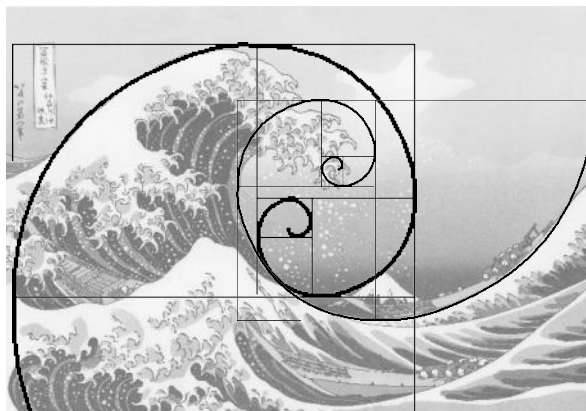
ヴィーナス



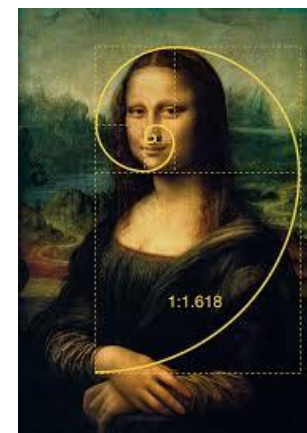
黄金螺旋



神奈川冲浪裏(葛飾北斎)



モナ・リザ(ダヴィンチ)



# 研究背景



自動車エレクトロニクスに注目が集まる



マイクロコントローラーには  
従来よりも**高速**かつ**高信頼性**のSAR ADCが必要に

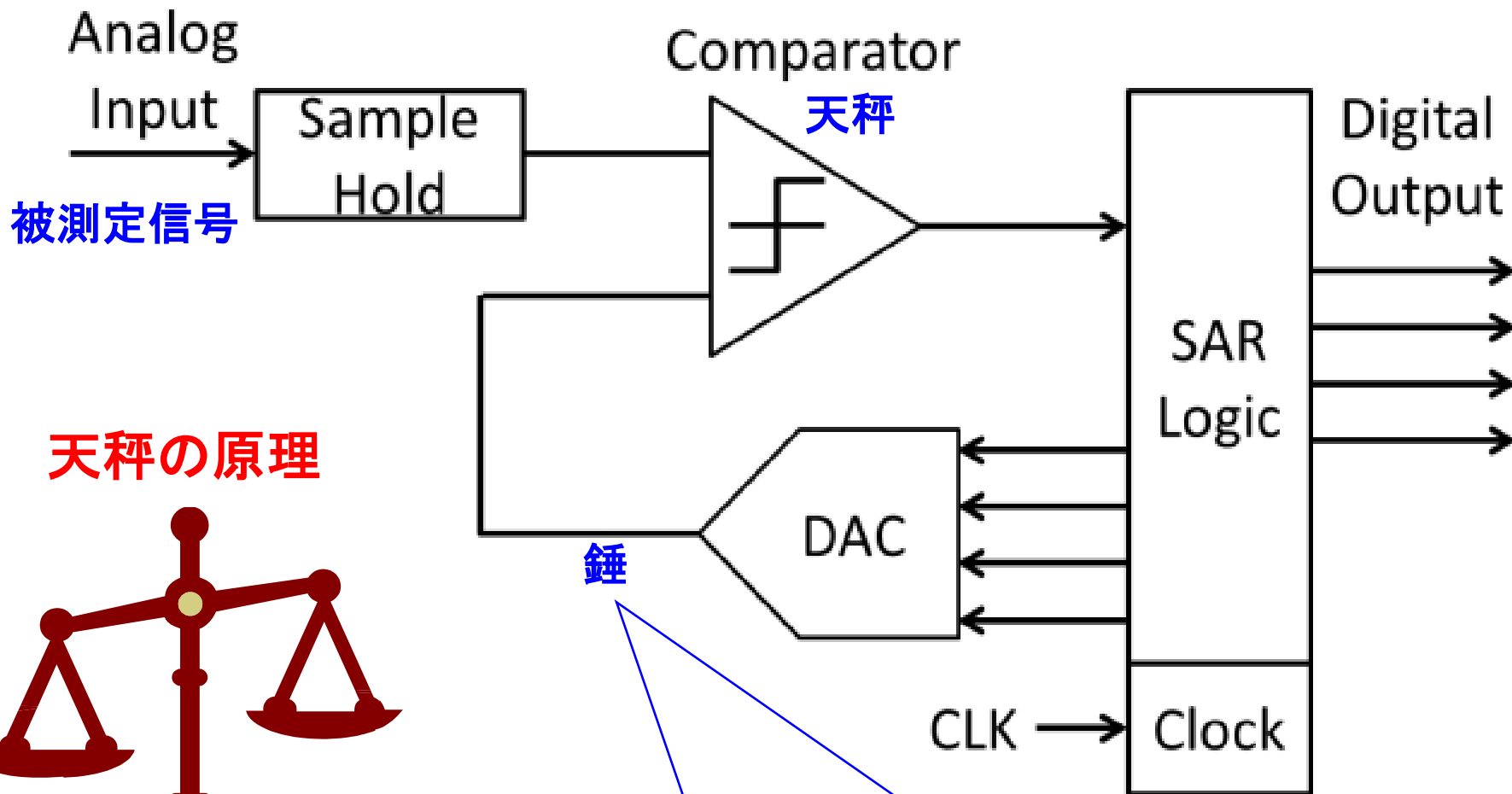


デジタル誤差補正可能な冗長設計で解決！

設計課題の存在 (☹)

SAR ADC: Successive Approximation Register Analog-to-Digital Converter

# 逐次比較近似AD変換器



天秤の原理



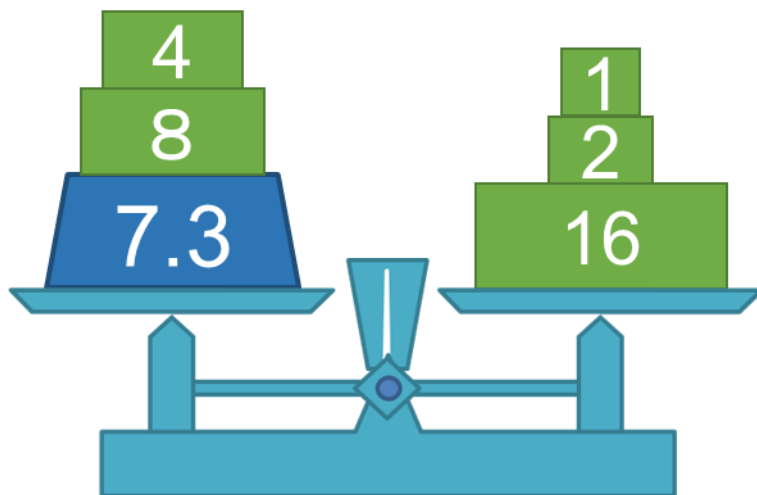
一般的に 二進重みを利用  
(1, 2, 4, 8, 16, 32, 64 ...)



# SAR ADCの二進探索動作(1)

## 5bit5step SAR ADC

- 二進重み
  - $p(k) = 16, 8, 4, 2, 1$
- 入力 7.3 LSB 相当



$$7.3 \Rightarrow 7$$

Step	1st	2nd	3rd	4th	5th	output
Weight $p(k)$	16	8	4	2	1	
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

## SAR ADCの二進探索動作(2)

5bit 5step SAR ADC

➤ 二進重み

$$p(k) = 16, 8, 4, 2, 1$$

➤ 入力 7.3 LSB 相当

二進数と十進数が1対1に対応

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

$$7 = 16 - 8 - 4 + 2 + 1 + 0.5 - 0.5$$

Step	1st	2nd	3rd	4th	5th	output
Weight p(k)	16	8	4	2	1	
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

Level

# SAR ADCの二進探索動作(3)

## 5bit5step SAR ADC

- 二進重み  
 $p(k) = 16, 8, 4, 2, 1$
- 入力 7.3 LSB 相当

二進数と十進数が1対1に対応

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



一回の判定誤りが  
直接出力間違いとなる

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

信頼性の劣化

Step	1st	2nd	3rd	4th	5th	output
Weight p(k)	16	8	4	2	1	
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	0	1	0	0	0	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						2
1						1
0						0

Level

# SAR ADCの冗長設計

冗長: 余分や余裕のこと

↓ SAR ADCへ適用

時間冗長性を利用

- ◆ 比較判定回数を増加
- ◆ 比較する電圧を変更

↓

デジタルコードによる  
表現方法増加

↓

デジタル誤差補正の実現

二進重み

$p(k): 1, 2, 4, 8, 16$



↓

非二進重み

$p(k): 1, 2, 3, 6, 10, 16$



# SAR ADCの冗長探索動作(1)

## 5bit6step SAR ADC

- 冗長重み  
 $p(k) = 16, 10, 6, 3, 2, 1$
- 入力 6.3 LSB 相当

判定ステップの増加

$6 \Rightarrow 010001 \Rightarrow 6$

$$16 - 10 + 6 - 3 - 2 - 1 + 0.5 - 0.5 = 6$$

Step	1st	2nd	3rd	4th	5th	6th	output
Weight p(k)	16	10	6	3	2	1	
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	0	1	0	0	0	1	3
2							2
1							1
0							0

Level



# SAR ADCの冗長探索動作(2)

## 5bit6step SAR ADC

- 冗長重み  
 $p(k) = 16, 10, 6, 3, 2, 1$
- 入力 6.3 [LSB]

判定ステップの増加

6 ⇒ 010001 ⇒ 6



デジタルコード表現が複数

6 ⇒ 001111 ⇒ 6

デジタル誤差補正

高信頼性の実現

Step	1st	2nd	3rd	4th	5th	6th	output
Weight p(k)	16	10	6	3	2	1	
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	0	0	1	1	1	1	3
2							2
1							1
0							0

Level

判定間違い!

# 補正力の評価方法

補正力を比較するための  
評価方法が必要

## 補正能力 $q(k)$ を定義

一回誤っても後段で補正できる  
入力電圧と比較電圧の絶対差

1<sup>st</sup> step 補正能力  
 $q(1)=3$

判定間違い

Step	1st	2nd	3rd	4th	5th	6th	output
Weight $p(k)$	16	10	6	3	2	1	
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							2
1							1
0							0

出力は18まで戻せる

# 補正できる入力範囲差 $q$

補正力を比較するための  
評価方法が必要

## 補正能力 $q(k)$ を定義

一回誤っても後段で補正できる  
入力電圧と比較電圧の絶対差

補正可能である条件

$$|V_{in} - V_{ref}(k)| \leq q(k)$$



$q(k)$ が大きい=補正力高い

Step	1st	2nd	3rd	4th	5th	6th	output
Weight $p(k)$	16	10	6	3	2	1	
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							2
1							1
0							0

Level

$q(1)$

補正可能な  
入力範囲

# 補正できる入力範囲差 $q$

補正力を比較するための  
評価方法が必要

$k$  step目で判定誤りがあっても  
後段で補正が可能な入力範囲差 $q(k)$

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

$p(k)$ :  $k$  step目の比較重み

$M$ : 総ステップ数

補正力は比較重み $p(k)$ によって決まる

どのような比較重みを利用するか？



Step	1st	2nd	3rd	4th	5th	6th	output
Weight $p(k)$	16	10	6	3	2	1	
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	↕	▲ $q(1)$					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							2
1							1
0			↑				0

# 比較電圧重み $p(k)$ の決定(従来手法)

$N$  bit 全  $M$  step 中  $k$  step 目の比較重み  $p(k)$  を決定

## 従来手法

① 基数radixから決定する  $\Rightarrow p(k) = r^{M-k} \quad (1 < r < 2)$

- 適切な基数の決定が難しい
  - 補正力(信頼性)と変換ステップ数(時間)はトレードオフ
- $p(k)$  は必ず小数になる(単位セルによる実現困難)
  - 面積比を基数比で利用  $\Rightarrow$  精度が悪い
  - 四捨五入して整数を使う  $\Rightarrow$   $q(k)$  のばらつき

② 最も適当な $p(k)$ を任意に決定する

- 適切な効果を得づらい
- 決定が難しく設計時間を増加

# 従来手法の問題点

## 5bit 6step SAR ADC

### 冗長設計手法①

radix=1.80

比較電圧重み $p(k)$

$$p(1) = 2^{5-1} = 16$$

$$p(2) = 1.8^4 \cong 10$$

$$p(3) = 1.8^3 \cong 6$$

$$p(4) = 1.8^2 \cong 3$$

$$p(5) = 1.8^1 \cong 2$$

$$p(6) = 1.8^0 = 1$$

絶対に補正できない入力範囲

冗長設計効果の減少



適切な $p(k)$ 選択手法が重要

Step	1st	2nd	3rd	4th	5th	6th	output
Weight $p(k)$	16	10	6	3	2	1	
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	↕	▲ $q(1)$					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							2
1							1
0			↑				0

# フィボナッチ数列とは (再掲)

## フィボナッチ数列

$$F_0 = 0$$

$$F_1 = 1$$

$$F_{n+2} = F_n + F_{n+1}$$

## 初めの項を計算

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...

⇒ フィボナッチ数と呼ばれる



Leonardo Fibonacci  
(伊: 1170-1250年頃)

また隣り合う項の比率は以下の値 $\varphi$ に収束

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

収束比率  $\varphi$  : 黄金比 (Golden ratio)

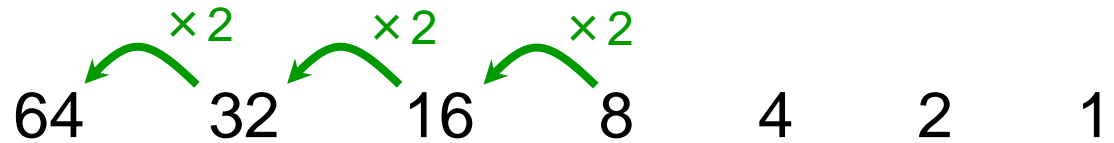
# 比較電圧重み $p(k)$ の決定(提案手法)

$N$  bit 全  $M$  step 中  $k$  step 目の比較重み  $p(k)$  を決定

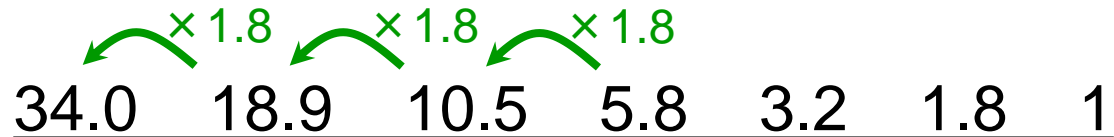
提案手法

フィボナッチ数を $p(k)$ として利用する  $\Rightarrow p(k) = F_{M-k+1}$

Binary Weight  
二進数



Radix 1.8 Weight  
1.8進数



Fibonacci Weight  
約1.62進数



隣り合う項の比率が黄金比 $\varphi$ に収束

$\Rightarrow$  整数のみで約1.62進数 ( $radix = 1.62$ ) を実現できる!



# フィボナッチ数列を用いたSAR ADC

## フィボナッチ数列SAR ADC

2点の性質を新発見！

- ① 許容値 $q(k)$ は必ずフィボナッチ数
- ② 許容できる範囲が必ず接する

Step	1st	2nd	3rd	4th	5th	6th	7th
Weight $p(k)$	16	8	5	3	2	1	1
33					↓		
32				↕			
31				↕			
30			↕		↕		
29			↕		↕		
28			↕		↕		
27			↕		↕		
26		↕		↕			
25		↕		↕			
24		↕		↕			
23		↕		↕			
22		↕		↕			
21		↕		↕			
20	↕	↕	↕	↕			
19	↕	↕	↕	↕			
18	↕	↕	↕	↕			
17	↕	↕	↕	↕			
16	↕	↕	↕	↕			
15	↕	↕	↕	↕			
14	↕	↕	↕	↕			
13	↕	↕	↕	↕			
12	↕	↕	↕	↕			
11	↕	↕	↕	↕			
10	↕	↕	↕	↕			
9	↕	↕	↕	↕			
8	↕	↕	↕	↕			
7	↕	↕	↕	↕			
6	↕	↕	↕	↕			
5	↕	↕	↕	↕			
4	↕	↕	↕	↕			
3	↕	↕	↕	↕			
2	↕	↕	↕	↕			
1	↕	↕	↕	↕			
0	↕	↕	↕	↕			
-1	↕	↕	↕	↕			
-2	↕	↕	↕	↕			

Level

# 性質②の意義

## フィボナッチ数列SAR ADC

2点の性質を新発見！

- ① 許容値 $q(k)$ は必ずフィボナッチ数
- ② 許容できる範囲が必ず接する



フィボナッチ数重み( $r = 1.618$ )より...

- ・基数  $r$  が大きい  $\Rightarrow$   $q(k)$  は離れる
- ・基数  $r$  が小さい  $\Rightarrow$   $q(k)$  は重なる



黄金比 $\phi$ は  
冗長さの境界条件であり  
設計指針となる

Step	1st	2nd	3rd	4th	5th	6th	7th
Weight p(k)	16	8	5	3	2	1	1
33					↓		
32				↕			
31				↕			
30			↕		↕		
29			↕		↕		
28			↕		↕		
27			↕		↕		
26		↕		↕			
25		↕		↕			
24		↕		↕			
23		↕		↕			
22		↕		↕			
21		↕		↕			
20	↕	↕		↕			
19	↕	↕		↕			
18	↕	↕		↕			
17	↕	↕		↕			
16	↕	↕		↕			
15	↕	↕		↕			
14	↕	↕		↕			
13	↕	↕		↕			
12	↕	↕		↕			
11	↕	↕		↕			
10	↕	↕		↕			
9	↕	↕		↕			
8	↕	↕		↕			
7	↕	↕		↕			
6	↕	↕		↕			
5	↕	↕		↕			
4	↕	↕		↕			
3	↕	↕		↕			
2	↕	↕		↕			
1	↕	↕		↕			
0	↕	↕		↕			
-1	↕	↕		↕			
-2	↕	↕		↕			

Level

Red arrows indicate  $q(1)$  at level 16,  $q(2)$  at level 8,  $q(3)$  at level 4,  $q(4)$  at level 6, and  $q(5)$  at level 7.

# 性質②の意義

## フィボナッチ数列SAR ADC

2点の性質を新発見！

- ① 許容値 $q(k)$ は必ずフィボナッチ数
- ② 許容できる範囲が必ず接する



接する境界で  
すべての入力範囲をもれなく  
カバーすることになる



黄金比 $\phi$ を使うことで

- ・無駄なステップ
- ・補正できない入力範囲

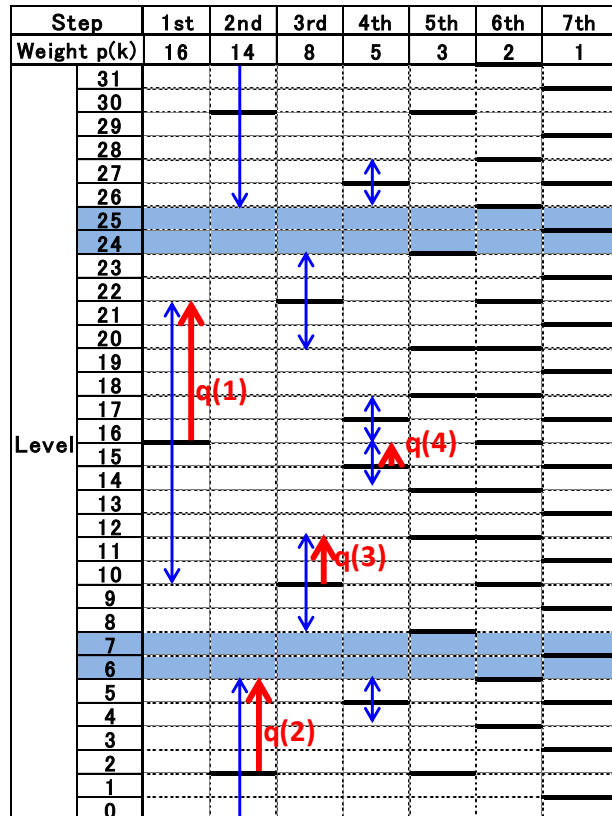
がない最も効率のよい設計が可能

Step	1st	2nd	3rd	4th	5th	6th	7th
Weight p(k)	16	8	5	3	2	1	1
33					↓		
32				↕			
31				↕			
30			↕		↕		
29			↕		↕		
28			↕		↕		
27			↕		↕		
26		↕		↕			
25		↕		↕			
24		↕		↕			
23		↕		↕			
22		↕		↕			
21		↕		↕			
20	↕	↕		↕			
19	↕	↕		↕			
18	↕	↕		↕			
17	↕	↕		↕			
16	↕	↕		↕			
15	↕	↕		↕			
14	↕	↕		↕			
13	↕	↕		↕			
12	↕	↕		↕			
11	↕	↕		↕			
10	↕	↕		↕			
9	↕	↕		↕			
8	↕	↕		↕			
7	↕	↕		↕			
6	↕	↕		↕			
5	↕	↕		↕			
4	↕	↕		↕			
3	↕	↕		↕			
2	↕	↕		↕			
1	↕	↕		↕			
0	↕	↕		↕			
-1	↕	↕		↕			
-2	↕	↕		↕			

# 従来手法との比較(5bit ADC)

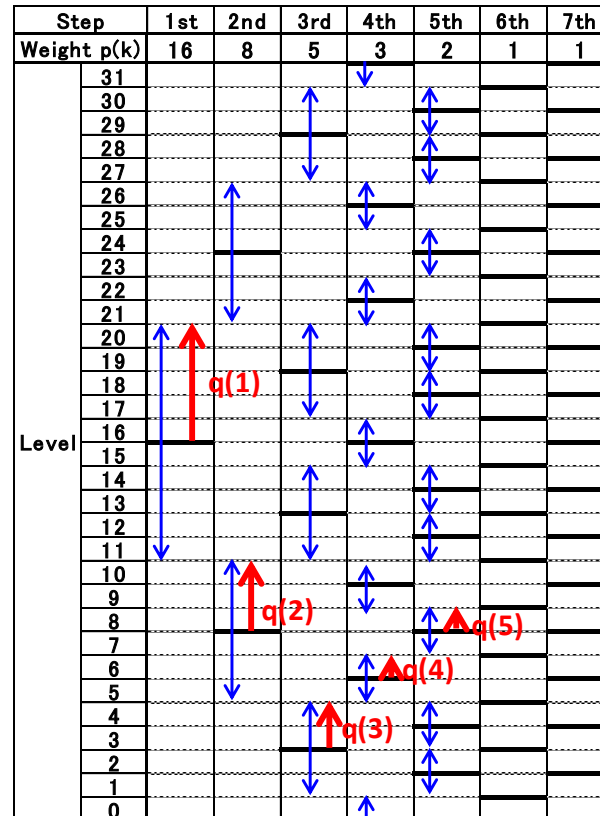
従来手法

1.70進数



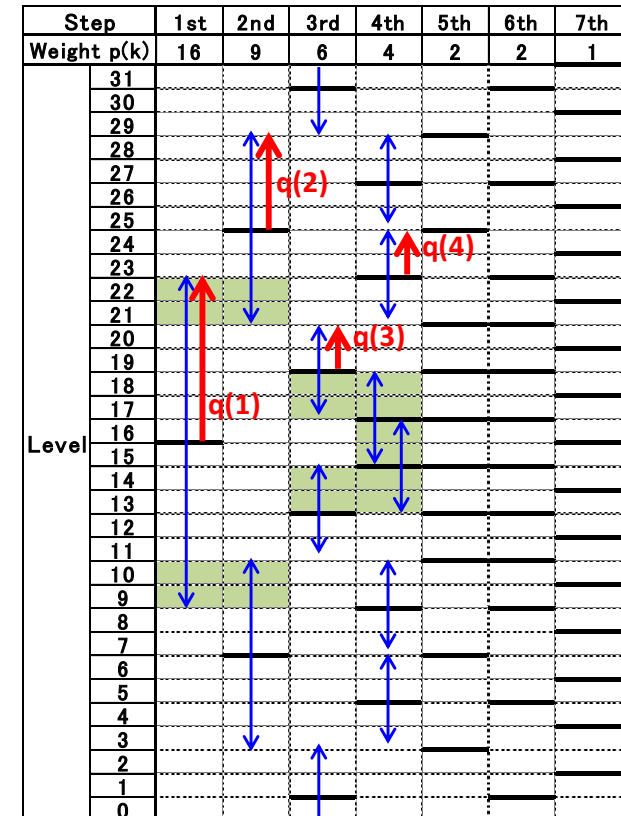
提案手法

1.62進数



従来手法

1.55進数

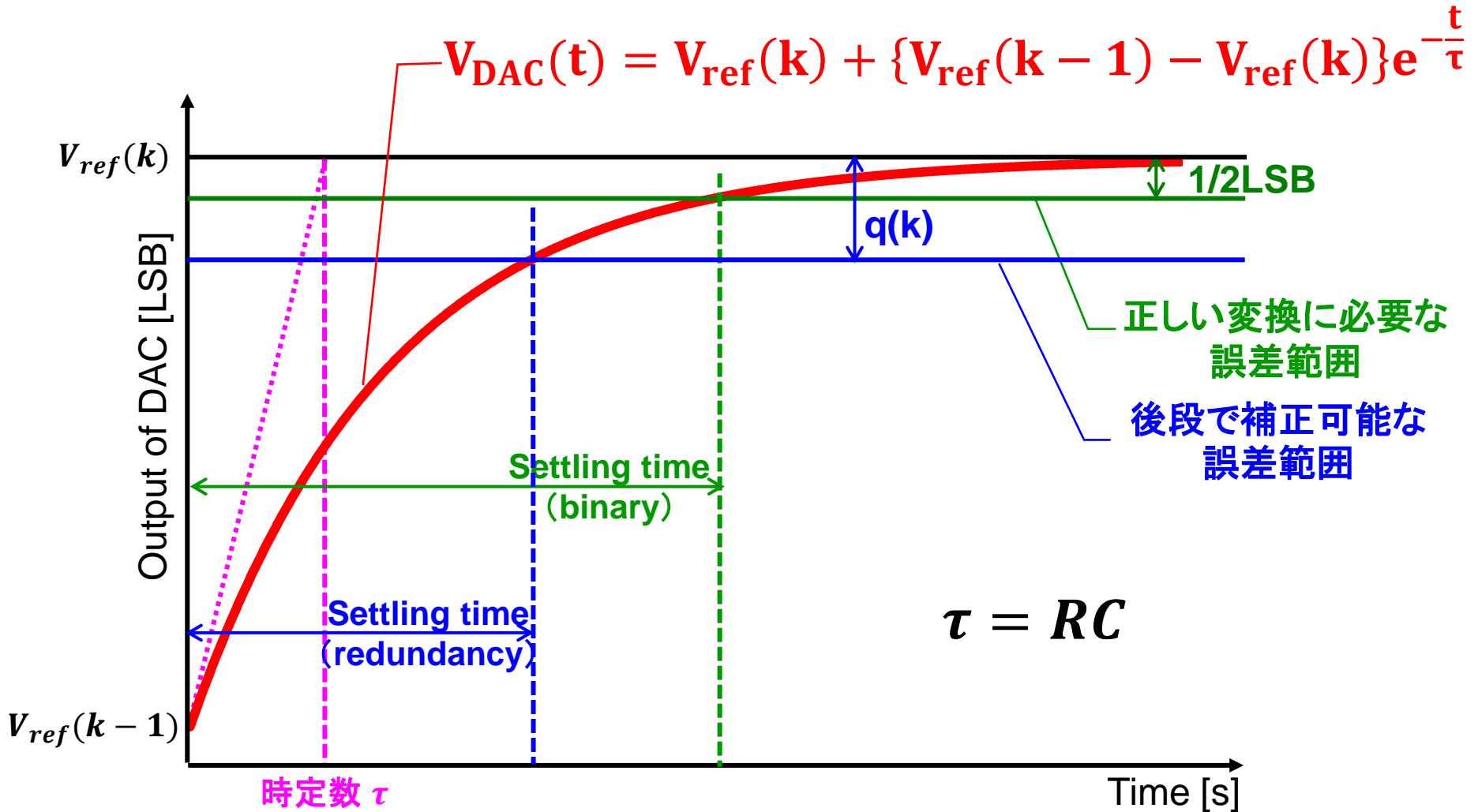


フィボナッチ数列冗長手法

冗長基数の境界条件  
効率の良い基準重み

# DAC出力の整定時間

RCモデルでDAC出力の整定を考える

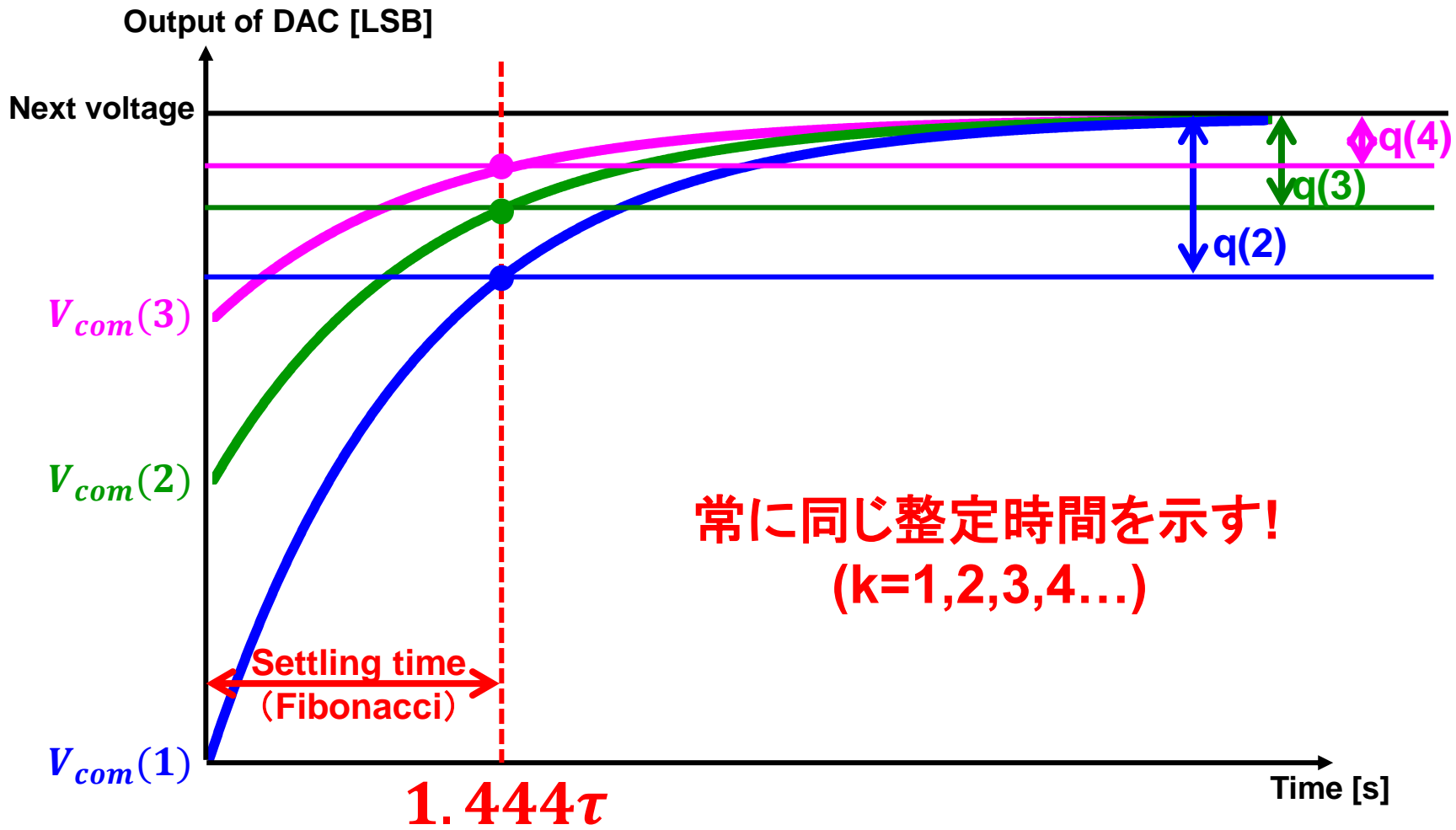


# フィボナッチ手法の整定時間の性質

新たな性質の発見！

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

$$= 1.444\tau \quad \text{for all } k$$

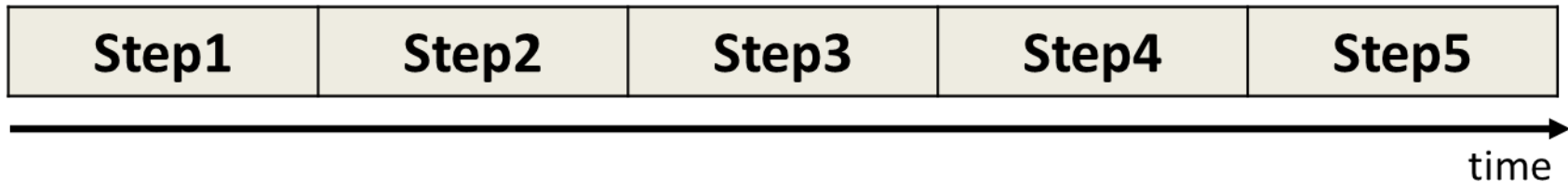


# 最短SAR AD変換時間

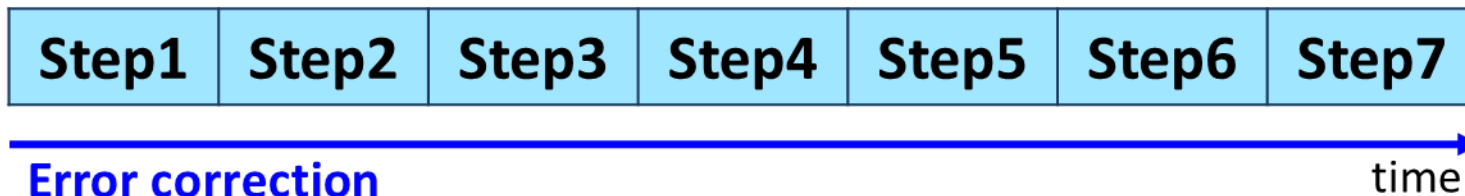
デジタル誤差補正 → DAC 不完全整定に対処

5bit SAR ADC

Binary search (complete settling)

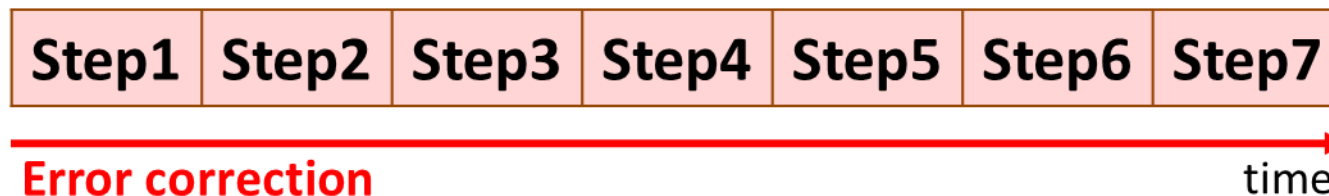


Redundant search (incomplete settling)



Error correction

Fibonacci search (incomplete settling)



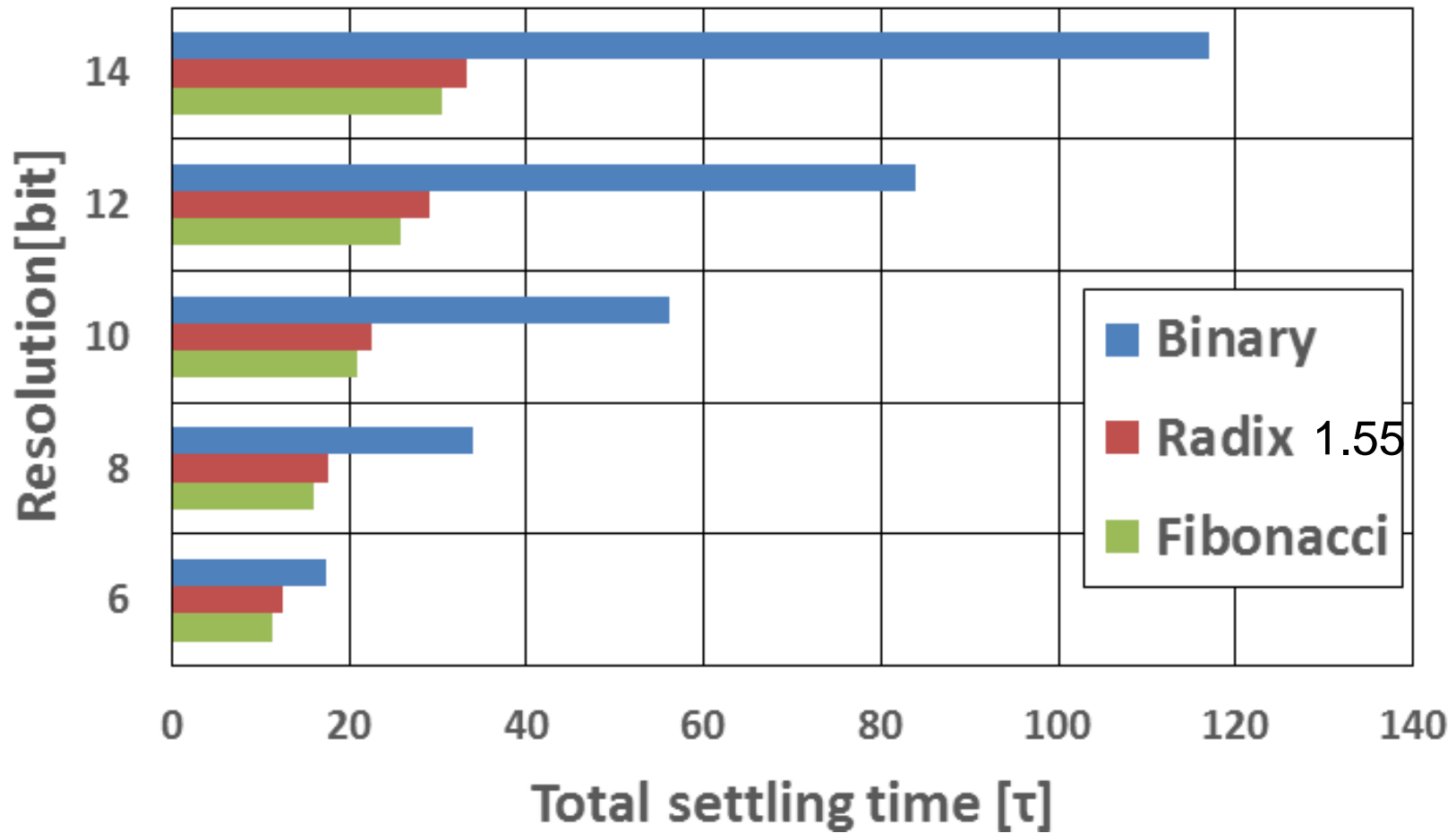
Error correction



最短  
AD変換時間  
の実現！

# AD変換時間 結果のまとめ

## 固定クロックにおける変換時間の比較(分解能別)



固定クロックでは.....

**どの分解能でもフィボナッチ手法SAR ADCが最も早い**



# フィボナッチ SAR ADC の結論

- 冗長SAR ADCの設計手法を提案
- フィボナッチ数列を利用することで重要な性質を得た
  - **高信頼性の実現**  
広い入力範囲を補正できる
  - **最速のSAR AD変換**  
固定クロック、DAC不完全整定の条件下
  - **基数の基準**  
黄金比 $\varphi (= 1.618)$ は基数の基準
  - **一定整定時間**  
DAC出力の要求整定時間はすべてのステップで一定

# セグメント型DA変換器の 魔方陣レイアウト技術による線形性向上

東野将史 小林春夫



# What's "Magic Square"?

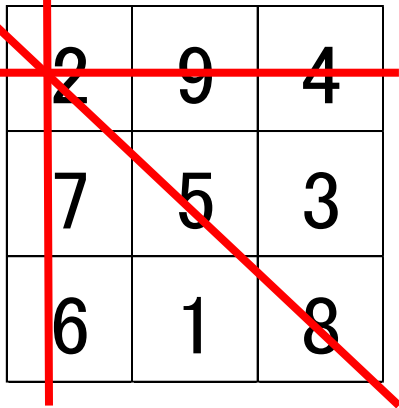


$$\sqrt[3]{16x^2}$$

# 魔方陣について

Magic square : 魔方陣

定和性 : 各行・列・対角線の和が一定



2	9	4
7	5	3
6	1	8

3次方陣

7	12	1	14
2	13	8	11
16	3	10	5
9	6	15	4

4次方陣

11	18	25	2	9
10	12	19	21	3
4	6	13	20	22
23	5	7	14	16
17	24	1	8	15

5次方陣

# 様々な魔方陣

## 完全魔方陣

7	12	1	14
2	13	8	11
16	3	10	5
9	6	15	4

10	5	3	16
15	4	6	9
8	11	13	2
1	14	12	7

## 対称魔方陣

## サイの目方陣

	○		○	○	○		○		
			○	○	○	○		○	
	○		○	○	○		○		
○	○	○	○		○				
	○			○		○	○	○	
○	○	○	○		○				
○		○					○	○	○
○		○		○		○		○	
○		○					○	○	○

# 様々な魔方陣(続き 1)

## ✓ 同心魔方陣

外側からひと側ずつ取り除いても、定和性を失わない



59	5	4	62	63	1	8	58
9	18	17	49	50	42	19	56
55	20	28	33	29	40	45	10
54	44	38	31	35	26	21	11
12	43	39	30	34	27	22	53
13	24	25	36	32	37	41	52
51	46	48	16	15	23	47	14
7	60	61	3	2	64	57	6

# 様々な魔方陣(続き 2)

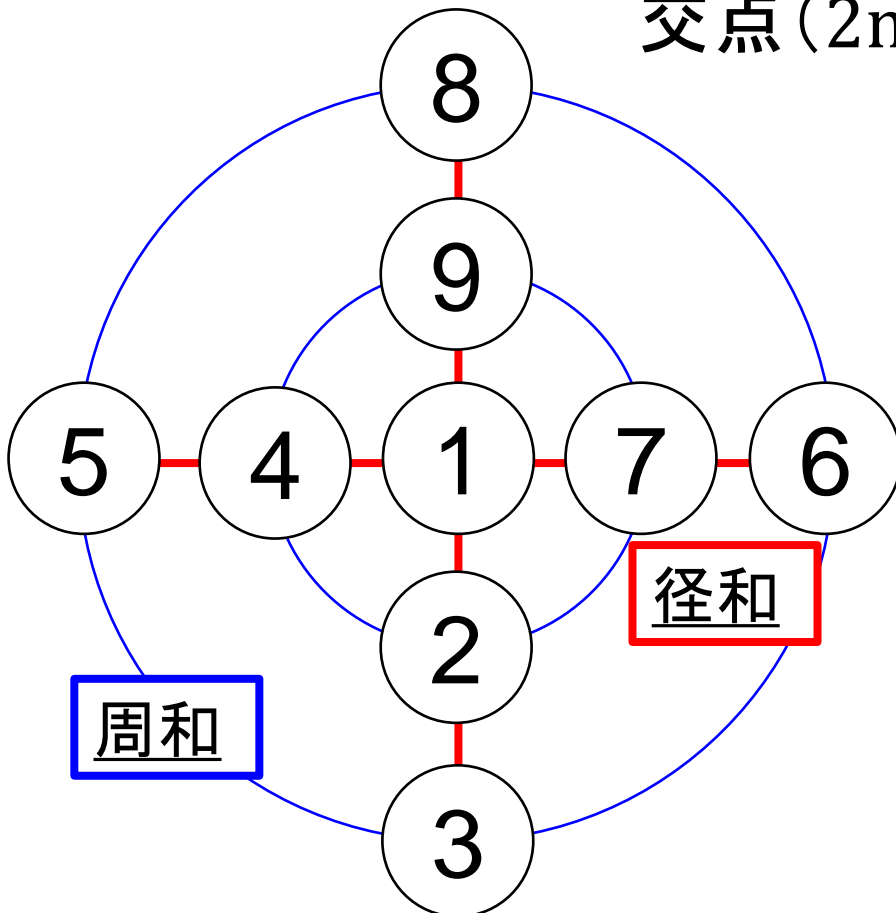
魔円陣：同心円と直径とを同じ個数だけ書き  
交点( $2n^2 + 1$ 個)に数字を置いたもの

$n=5$ (左図)

- 径和：直径上の $2n+1$ 個の和
- 周和：円周上の $2n$ 個の数と中心数の $2n+1$ 個の和

→ 径和と周和の一致

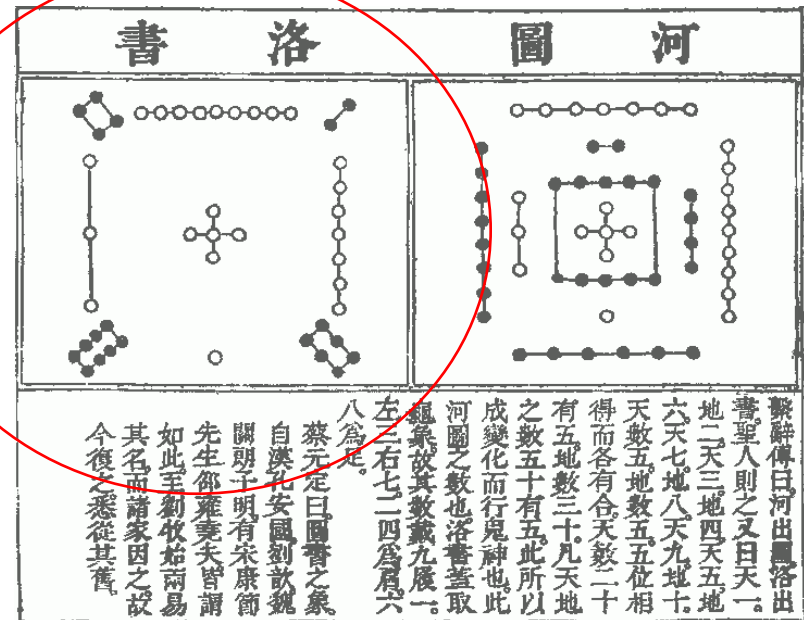
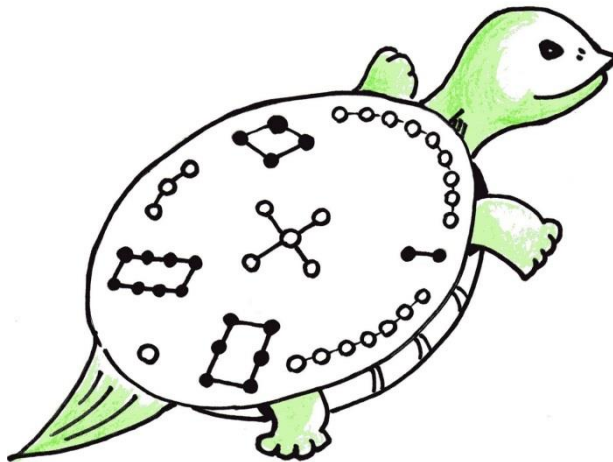
魔星陣, 立体魔方陣, etc,,,,,



# 魔方陣の歴史（中国）

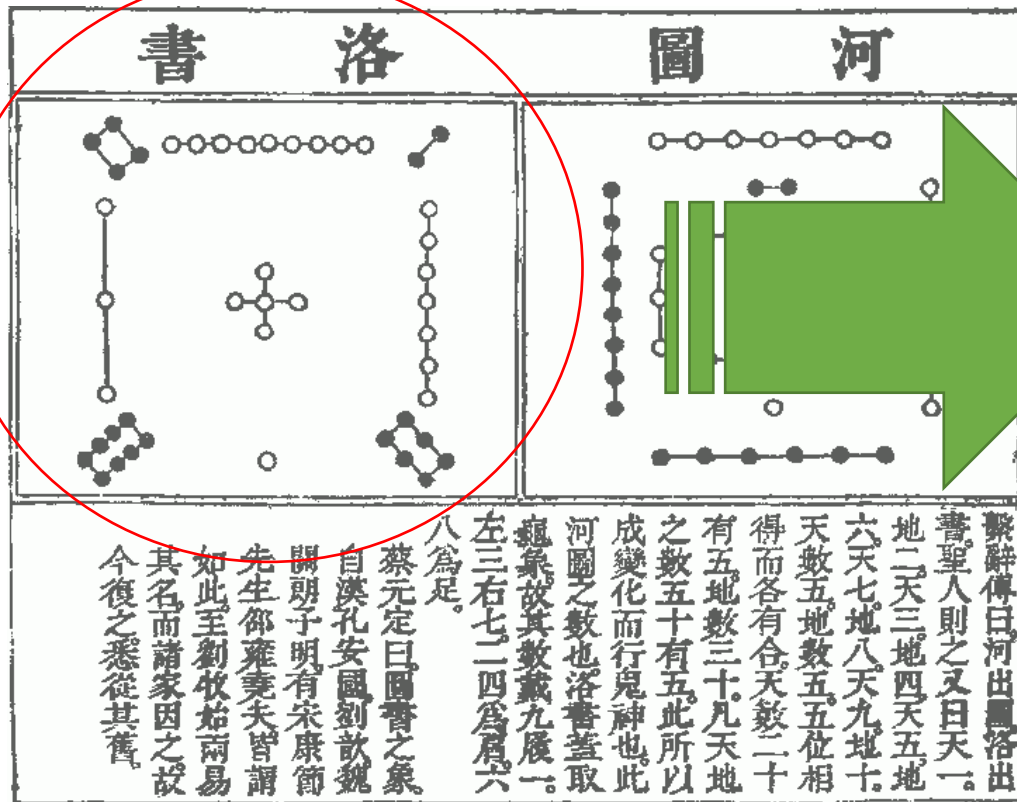
中国 紀元前

「夏(か)の禹王(うおう)が黄河の洪水を治めたとき、洛水から出た神亀の背に洛書が記されていた」





# 魔方陣の歴史（中国）



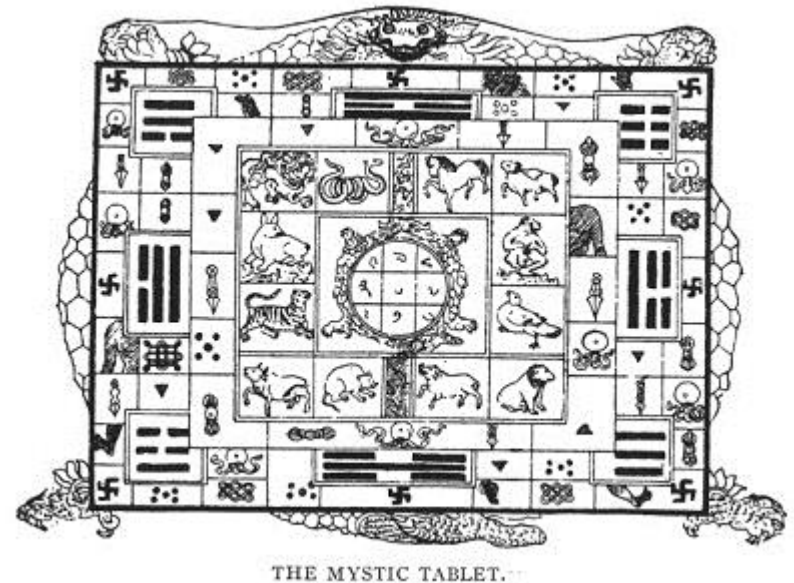
## 3次方陣

4	9	2
3	5	7
8	1	6

特殊な図であることから、  
九星術の根本として占星家が使用

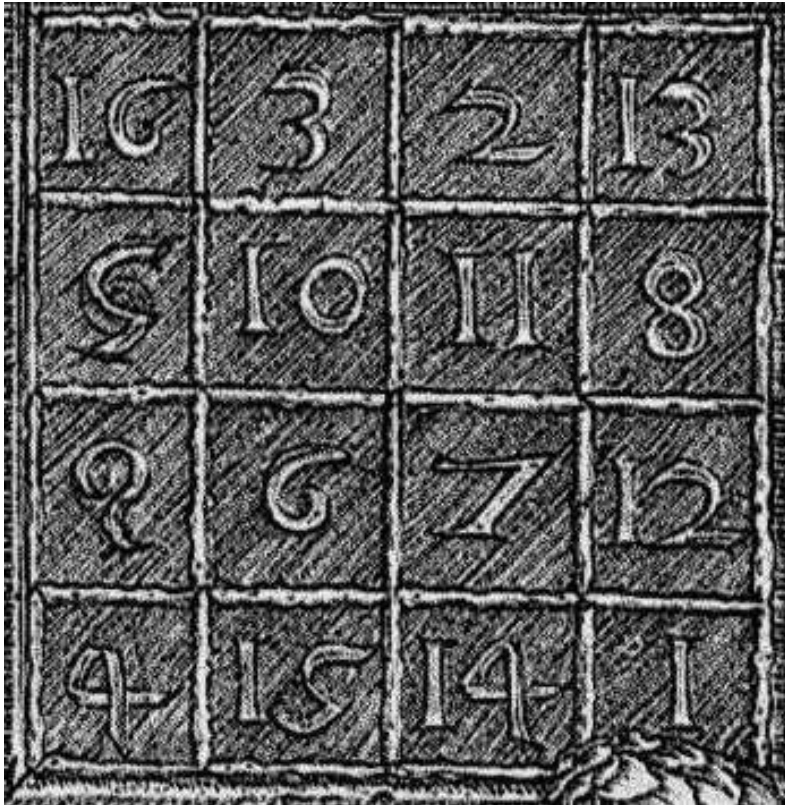
# 魔方陣の歴史（チベット、ネパール、ブータン）

中央の3次方陣の周りに十二支の動物を配した  
“生物の輪”を刻んだお守り

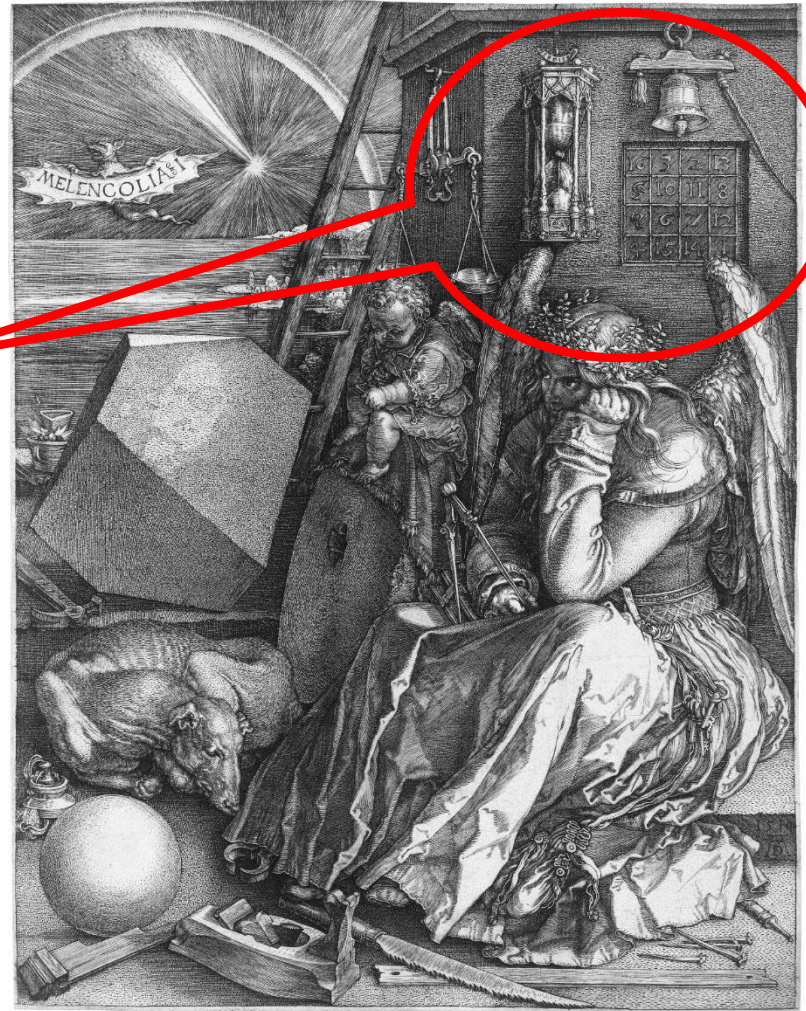


“ The mystic tablet “

15世紀 西洋



魔方陣



独: Melencolia I (1514)  
作: Albrecht Durer

# 魔方陣の歴史（日本）

## 上毛かるた

「和算の大家 関孝和」

関 孝和

日本数学史上最高の英雄人物

- 江戸時代の数学者
- 群馬県藤岡市出身
- 円周率の近似値，  
行列の概念を確立
- 魔方陣の研究「方陣之法」



# 余談

## 群馬県出身の学生

「群馬の若者は皆、子供の時の「上毛かるた」で関孝和を知っています。」

→ 学校教育の影響の大きさを認識

「すごいな。」

「じゃ国定忠治を知っているかい？」

「知りません。だれですか。」

「群馬には JR国定駅、赤城山があるけど。。。」

→ 若者に取り残される寂しさを感じる



# 魔方陣の歴史 (日本)

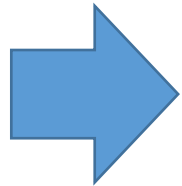
## 関孝和が考案した6次方陣

和が37となる2数を線で結ぶと模様が出現



4	3	35	36	28	5
6	14	19	15	26	31
30	24	17	21	12	7
29	25	16	20	13	8
10	11	22	18	23	27
32	34	2	1	9	33

和が37となる2数を線で結ぶと美しい模様が出現



37の連結線

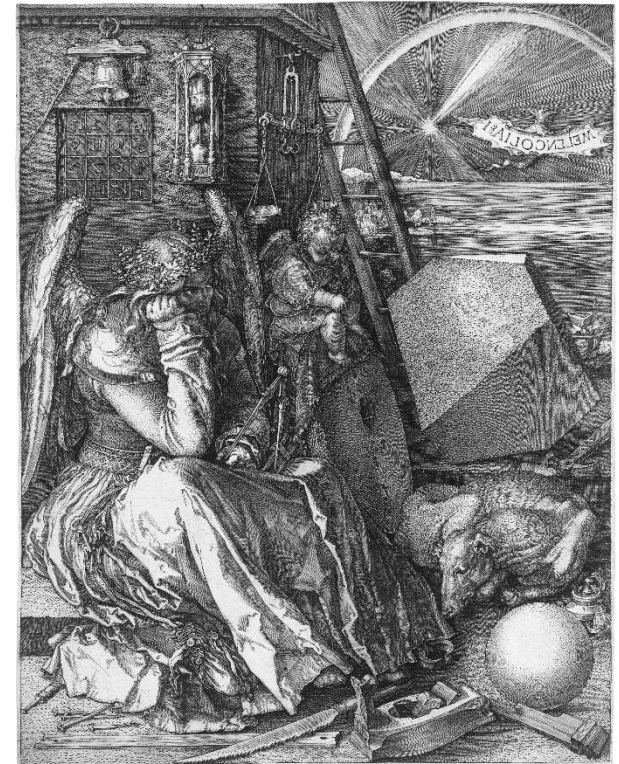
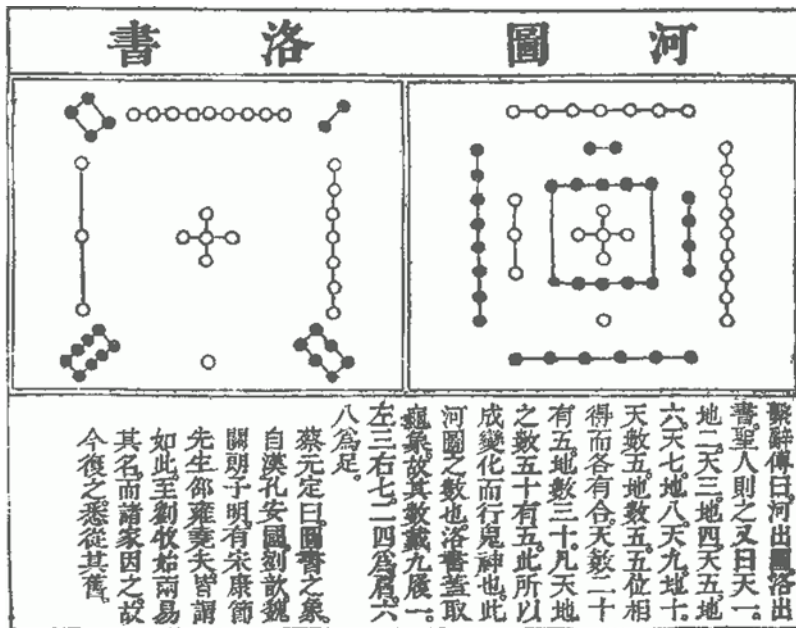
魔方陣の奥深さを感じる作品

<del>4</del>	<del>3</del>	<del>35</del>	<del>36</del>	<del>28</del>	<del>5</del>
<del>6</del>	<del>14</del>	<del>19</del>	<del>15</del>	<del>26</del>	<del>31</del>
<del>30</del>	<del>24</del>	17	21	<del>12</del>	<del>7</del>
<del>29</del>	<del>25</del>	16	20	<del>13</del>	<del>8</del>
<del>10</del>	<del>11</del>	<del>22</del>	<del>18</del>	<del>23</del>	<del>27</del>
<del>32</del>	<del>34</del>	<del>2</del>	<del>1</del>	<del>9</del>	<del>33</del>

# Magic Square – 魔方陣 -

## 「人類最初の数論問題」

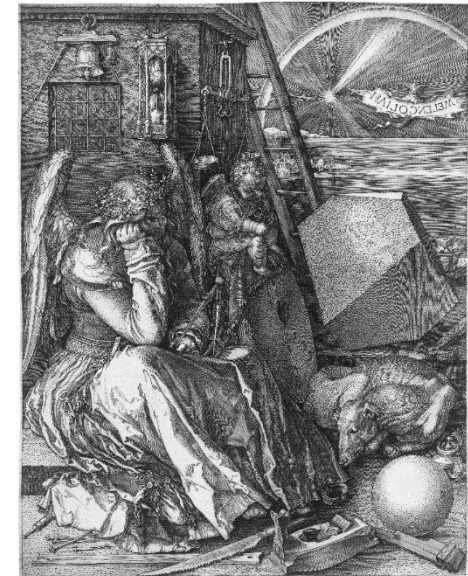
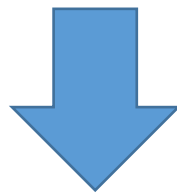
不思議な魔術ではなく、数の神秘が宿る





# Magic Square – 魔方陣 -

「人類最初の数論問題」を工学へ応用  
数の神秘の力より、イノベーション



魔方陣の「多様性」, 「調和」, 「奥深さ」, 「美しさ」

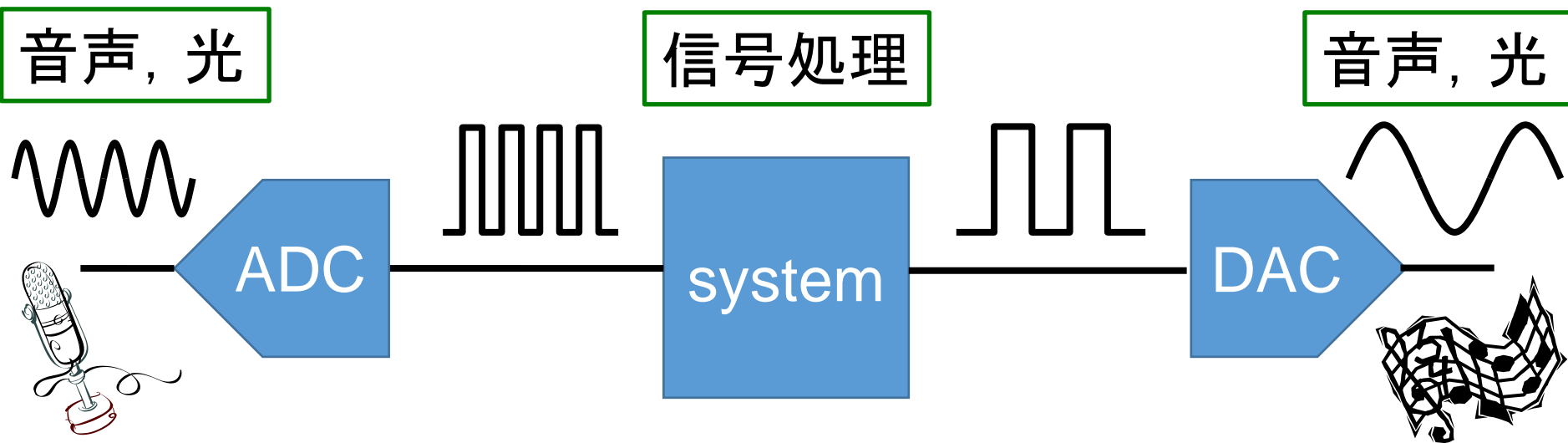
魔方陣レイアウト技術による  
DA変換器の線形性向上

# AD/DA変換器の重要性

電子機器 {  
• 小型化  
• 高速化

➡ デジタル回路が適している

✓ しかし、自然界の信号はアナログ信号であるので  
信号処理が必要



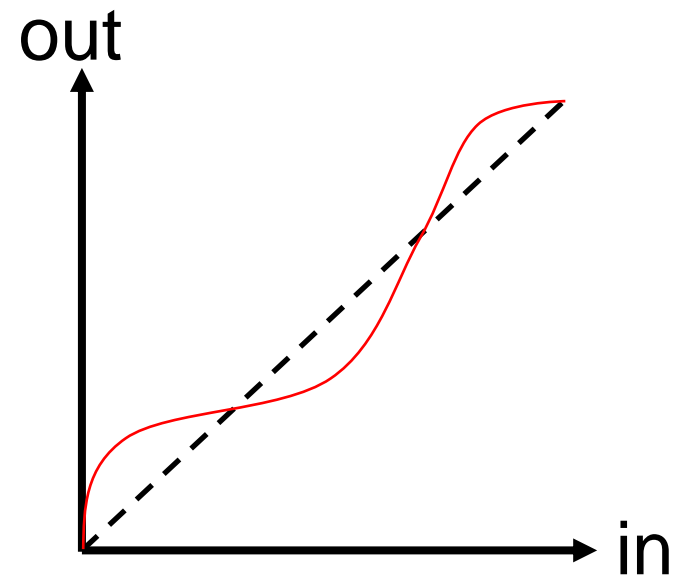
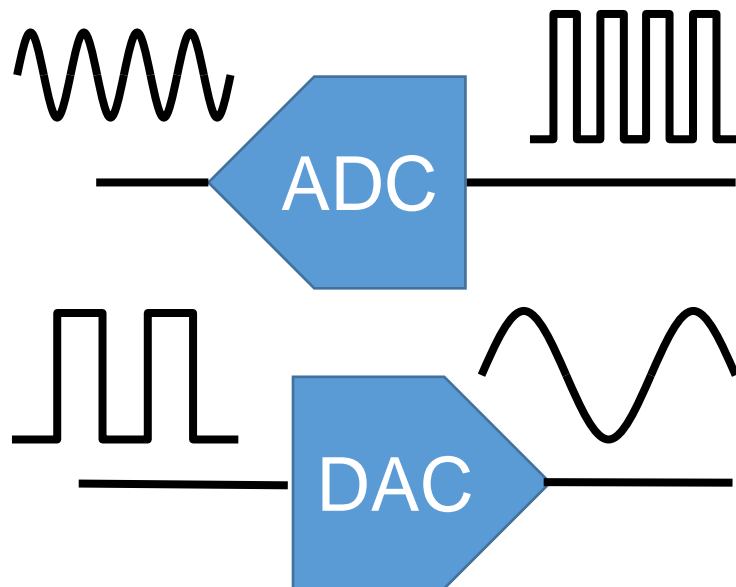
→ 高性能なAD及びDA変換器が求められている

# 素子ばらつきによる非線形性

- ✓ 半導体素子を構成しているシリコンウェハ上では、ばらつきが存在

ex) MOSFET特性, 抵抗, 容量

{ 素子のミスマッチ  
入出力信号の線形性劣化



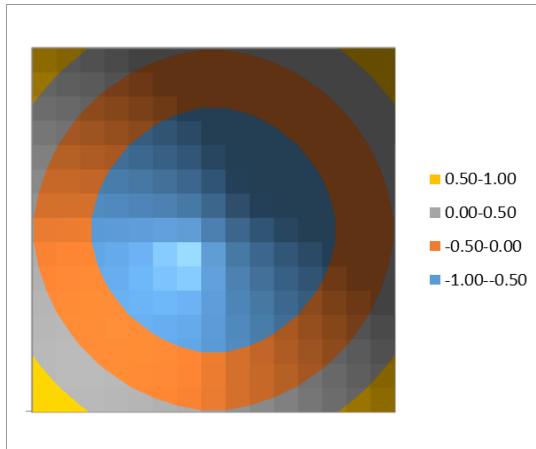
# 二種類の素子ばらつき

## ◆ ばらつきの種類

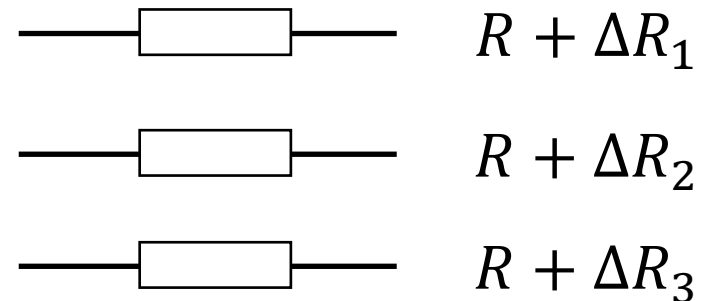
- ✓ システマティックなばらつき
- ✓ ランダムなばらつき

ex. ウェハ上で、

- システマティックな傾斜をもってばらつく
- 素子ごとにランダムにばらつく



システマティック



ランダム

# システムテックばらつき

## ◆ ばらつきの種類

- ✓ システムティックなばらつき
- ✓ ランダムなばらつき

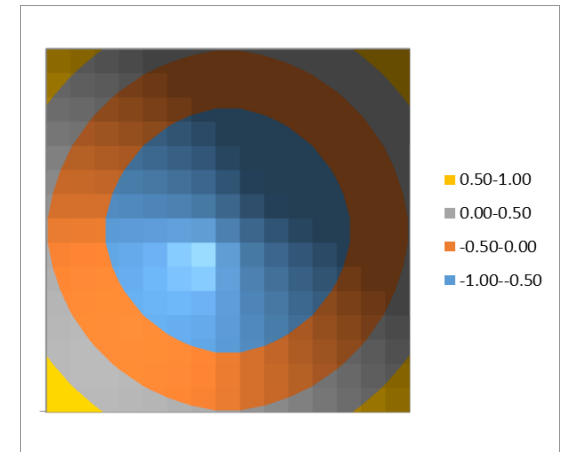
## ◆ システムティックなばらつき

- ✓ 電圧降下
- ✓ 酸化膜の厚さ
- ✓ ドーピング
- ✓ 機械的ストレス
- ✓ 温度分布
- ✓ ウエハ面内

### レイアウトで改善

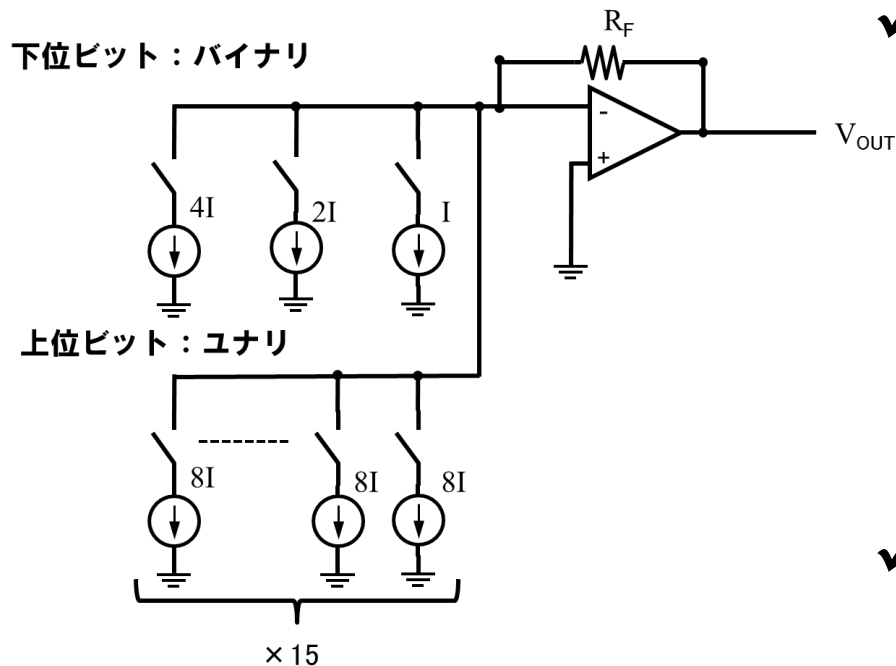
- 従来方法

Random Walk, 配置



# DA変換器の構成

## ◆ DA変換器の構成



セグメント型DA変換器

### ✓ バイナリ型

- 小型化可能
- コードの切り替えでグリッチ発生
- ミスマッチの発生：大

### ✓ ユナリ型

- 小型化不可
- ミスマッチの影響：少
- グリッチの発生：少

# DA変換器の動作

## ◆ DA変換器の構成

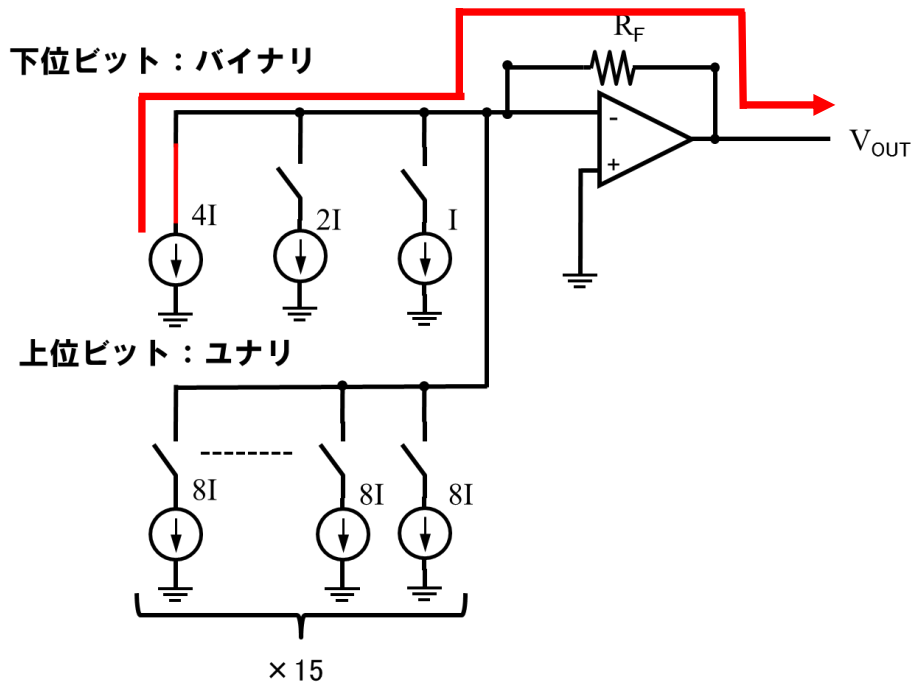
### ✓ 7bit DA変換器

ex.1

(0000100)



$$V_{out} = 4IR_F$$

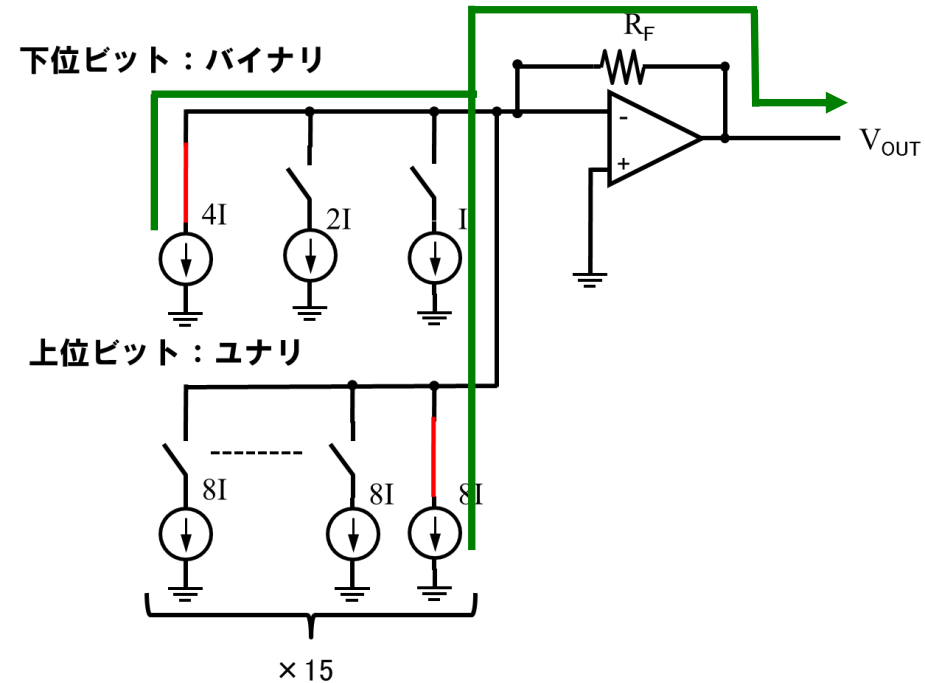


ex.2

(0001100)



$$V_{out} = 12IR_F$$



# 電流セル配列のレイアウト

## ◆ DA変換器の構成

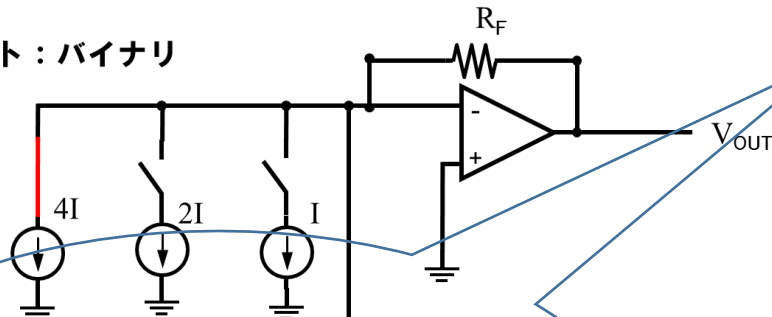
### ✓ 7bit DA変換器

(0001100)

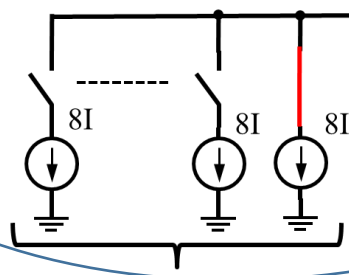


$$V_{out} = 12IR_F$$

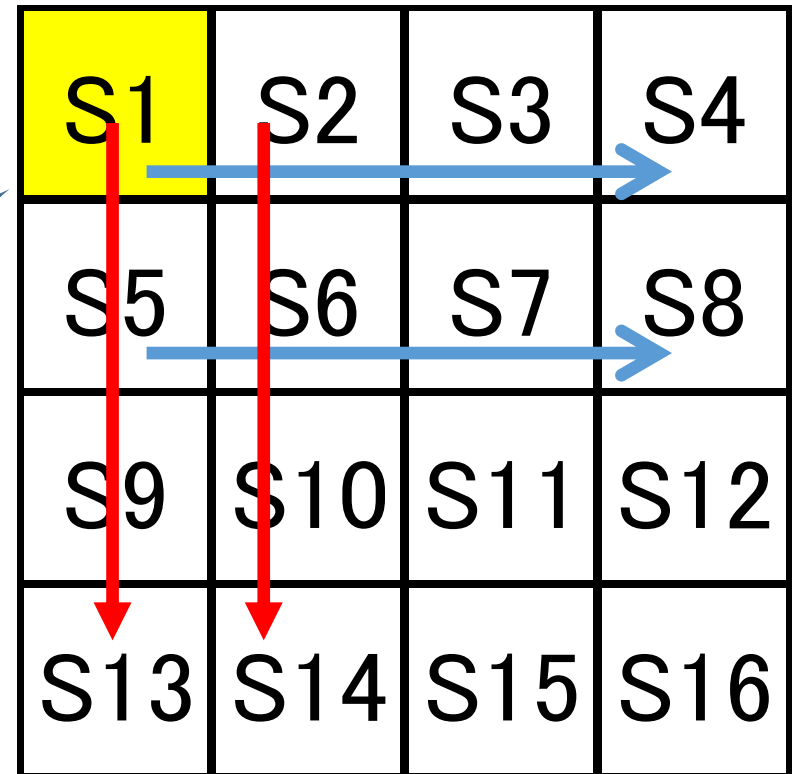
下位ビット：バイナリ



上位ビット：ユナリ



× 15

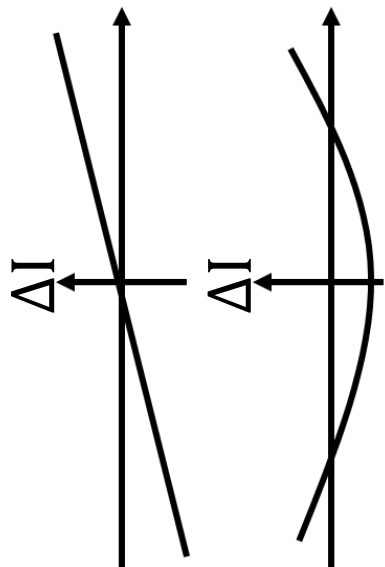
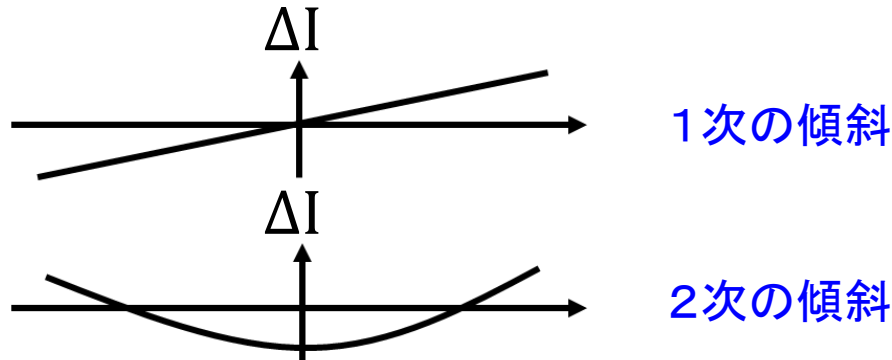


単位電流セル(ユナリ型)



# 電流源配列のミスマッチ傾斜

## ◆ DA変換器 - システムティック・ミスマッチとレイアウト

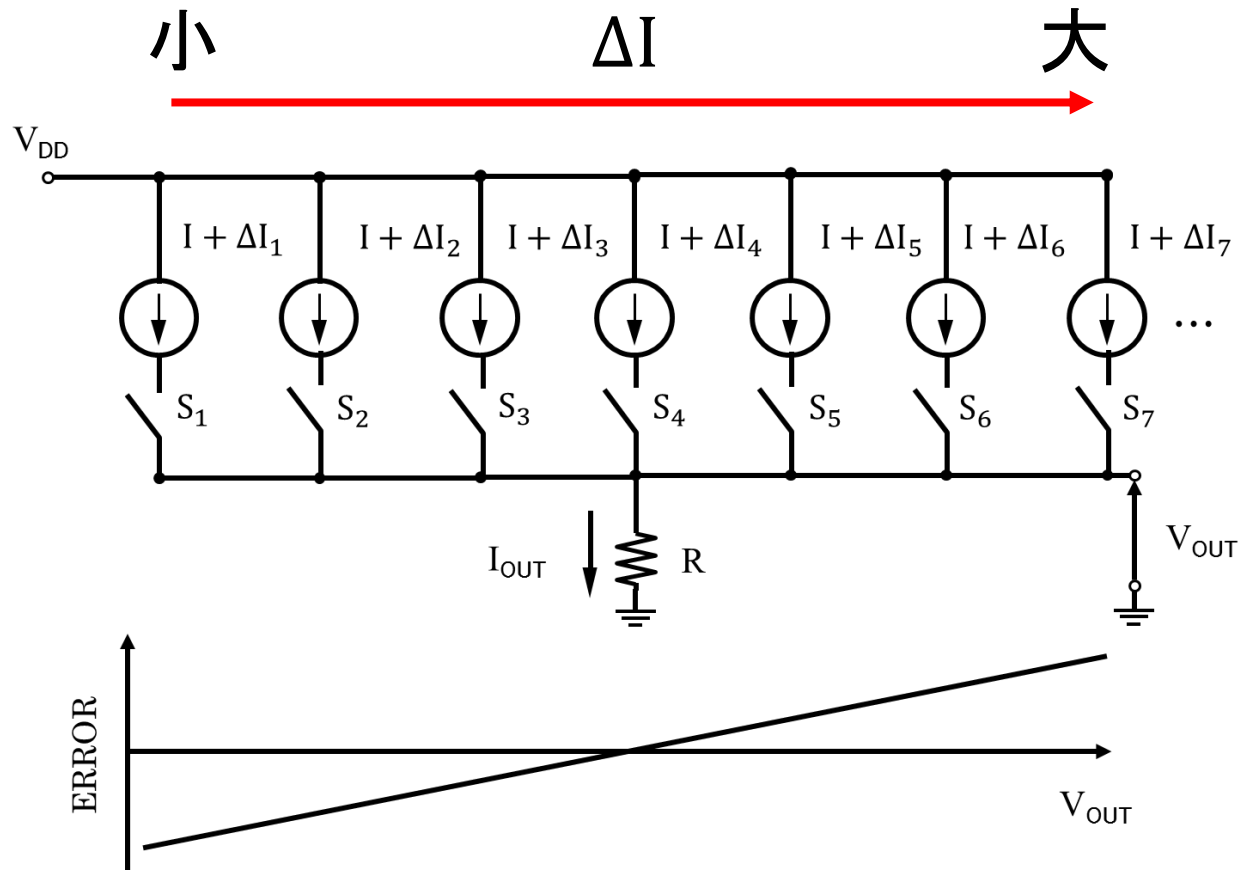


S1	S2	S3	S4
S5	S6	S7	S8
S9	S10	S11	S12
S13	S14	S15	S16

ミスマッチの傾斜がそのまま出力信号へ

# 電流セル配列とDAC非線形性

## ◆ DA変換器 - システムティック・ミスマッチとレイアウト

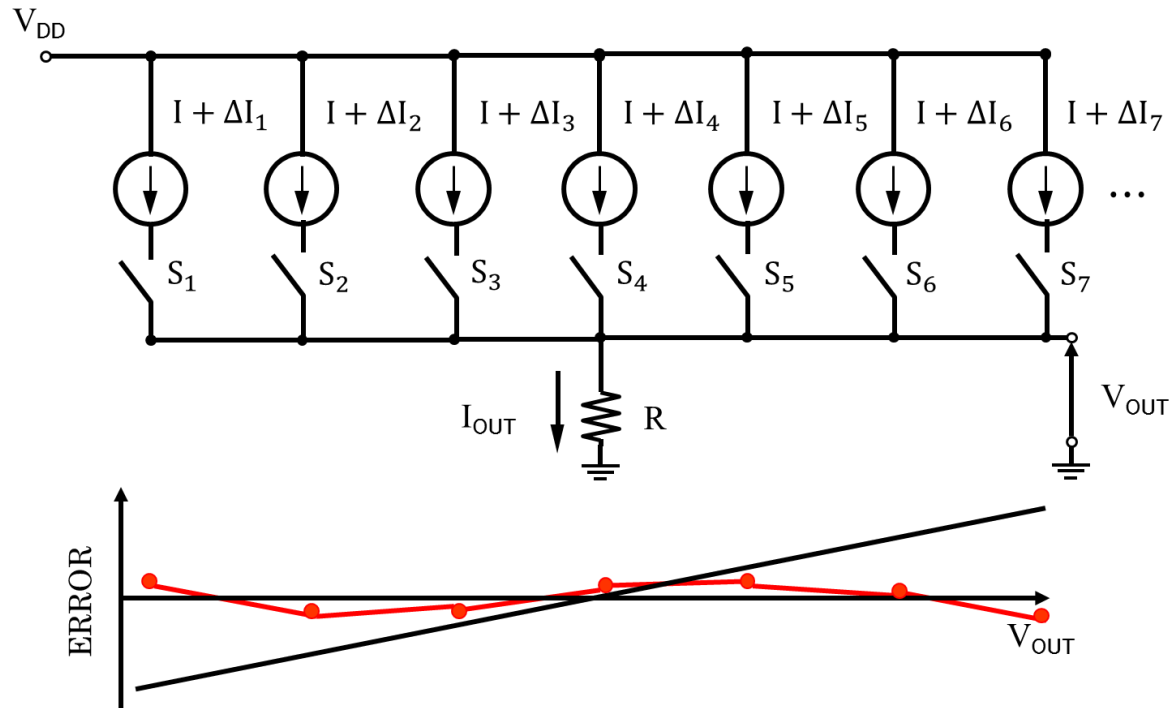


電流源のミスマッチにより入出力信号の線形性劣化が問題

# 電流セル配列レイアウトとDAC線形性向上

## ◆ DA変換器 - システムティック・ミスマッチとレイアウト

S4	S8	S12	S5
S14	S10	S6	S1
S9	S13	S2	S15
S3	S7	S16	S11



スイッチング順序を変える事によりエラーをキャンセル



Random Walk

# 魔方陣による電流セル配列レイアウト

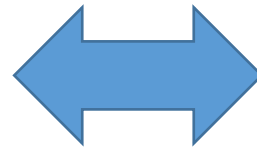
考案

魔方陣によるレイアウト

定和性の一致

4	9	7	14
16	5	11	2
13	8	10	3
1	12	6	15

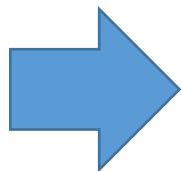
魔方陣



S1	S2	S3	S4
S5	S6	S7	S8
S9	S10	S11	S12
S13	S14	S15	S16

単位電流セル(ユナリ型)

魔方陣の「多様性」, 「調和」, 「奥深さ」, 「美しさ」



DA変換器への応用

## ✓ システムティックなばらつき

- Linear Error

$$\varepsilon_l(x, y) = g_l * \cos \theta * x + g_l * \sin \theta * y$$

$\theta$ : 傾きの角度,  $g_l$ : 傾きの大きさ

- Quadratic Error

$$\varepsilon_q(x, y) = g_q * (x^2 + y^2) - a_0$$

$g_q$ : 変化量,  $a_0$ : 位置

# 各魔方陣のシミュレーション方法・結果(2)

## ◆ 同心魔方陣

8次方陣を4つ組み合わせて8bitの単位電流源セルを表現

59	5	4	62	63	1	8	58
9	18	17	49	50	42	19	56
55	20	28	33	29	40	45	10
54	44	38	31	35	26	21	11
12	43	39	30	34	27	22	53
13	24	25	36	32	37	41	52
51	46	48	16	15	23	47	14
7	60	61	3	2	64	57	6

A1	B1
B2	A2

A:左図の魔方陣  
B:45° 左回転

# 各魔方陣のシミュレーション方法・結果(3)

## ◆ 同心魔方陣

- アルゴリズム

A1	B1
B2	A2

59	5	4	62	63	1	8	58	58	56	10	11	53	52	14	6
9	18	17	49	50	42	19	56	8	19	45	21	22	41	47	57
55	20	28	33	29	40	45	10	1	42	40	26	27	37	23	64
54	44	38	31	35	26	21	11	63	50	29	35	34	32	15	2
12	43	39	30	34	27	22	53	62	49	33	31	30	36	16	3
13	24	25	36	32	37	41	52	4	17	28	38	39	25	48	61
51	46	48	16	15	23	47	14	5	18	20	44	43	24	46	60
7	60	61	3	2	64	57	6	59	9	55	54	12	13	51	7
58	56	10	11	53	52	14	6	59	5	4	62	63	1	8	58
8	19	45	21	22	41	47	57	9	18	17	49	50	42	19	56
1	42	40	26	27	37	23	64	55	20	28	33	29	40	45	10
63	50	29	35	34	32	15	2	54	44	38	31	35	26	21	11
62	49	33	31	30	36	16	3	12	43	39	30	34	27	22	53
4	17	28	38	39	25	48	61	13	24	25	36	32	37	41	52
5	18	20	44	43	24	46	60	51	46	48	16	15	23	47	14
59	9	55	54	12	13	51	7	7	60	61	3	2	64	57	6

# 各魔方陣のシミュレーション方法・結果(4)

## ◆ 同心魔方陣

- アルゴリズム

A1	B1
B2	A2

59	5	4	62	63	1	8	58	58	56	10	11	53	52	14	6
9	18	17	49	50	42	19	56	8	19	45	21	22	41	47	57
55	20	28	33	29	40	45	10	1	42	40	26	27	37	23	64
54	44	38	31	35	26	21	11	63	50	29	35	34	32	15	2
12	43	39	30	34	27	22	53	62	49	33	31	30	36	16	3
13	24	25	36	32	37	41	52	4	17	28	38	39	25	48	61
51	46	48	16	15	23	47	14	5	18	20	44	43	24	46	60
7	60	61	3	2	64	57	6	59	9	55	54	12	13	51	7
58	56	10	11	53	52	14	6	59	5	4	62	63	1	8	58
8	19	45	21	22	41	47	57	9	18	17	49	50	42	19	56
1	42	40	26	27	37	23	64	55	20	28	33	29	40	45	10
63	50	29	35	34	32	15	2	54	44	38	31	35	26	21	11
62	49	33	31	30	36	16	3	12	43	39	30	34	27	22	53
4	17	28	38	39	25	48	61	13	24	25	36	32	37	41	52
5	18	20	44	43	24	46	60	51	46	48	16	15	23	47	14
59	9	55	54	12	13	51	7	7	60	61	3	2	64	57	6



# 各魔方陣のシミュレーション方法・結果(5)

## ◆ 同心魔方陣

- アルゴリズム

A1	B1
B2	A2

59	5	4	62	63	1	8	58	58	56	10	11	53	52	14	6
9	18	17	49	50	42	19	56	8	19	45	21	22	41	47	57
55	20	28	33	29	40	45	10	1	42	40	26	27	37	23	64
54	44	38	31	35	26	21	63	50	29	35	34	32	15	2	
12	43	39	30	34	27	22	63	62	49	33	31	30	36	16	3
13	24	25	36	32	37	41	52	4	17	28	38	39	25	48	61
51	46	48	16	15	23	47	14	5	18	20	44	43	24	46	60
7	60	61	3	2	64	57	6	59	9	55	54	12	13	51	7
58	56	10	11	53	52	14	6	59	5	4	62	63	1	8	58
8	19	45	21	22	41	47	57	9	18	20	44	43	24	46	60
1	42	40	26	27	37	23	64	55	20	28	33	29	40	45	10
63	50	29	35	34	32	15	2	54	44	38	31	35	26	21	11
62	49	33	31	30	36	16	3	12	43	39	30	34	27	22	53
4	17	28	38	39	25	48	61	13	24	25	36	32	37	41	52
5	18	20	44	43	24	46	60	51	46	48	16	15	23	47	14
59	9	55	54	12	13	51	7	7	60	61	3	2	64	57	6

# 各魔方陣のシミュレーション方法・結果(6)

## ◆ 同心魔方陣

- アルゴリズム

A1	B1
B2	A2

59	5	4	62	63	1	8	58	58	56	10	11	53	52	14	6
9	18	17	49	50	59	56	8	19	45	21	22	41	47	57	
55	20	29	33	29	40	45	10	1	42	40	26	27	37	23	64
54	44	38	31	35	26	21	63	50	29	35	34	32	15	2	
12	43	39	30	34	27	22	53	62	49	33	31	30	36	16	3
13	24	25	36	32	37	41	52	4	17	28	38	39	25	48	61
51	46	48	16	15	23	47	14	5	18	20	44	43	24	46	60
7	60	61	3	2	64	57	6	59	9	55	54	12	13	51	7
53	56	10	11	53	52	14	6	59	5	4	62	63	1	8	58
8	19	45	21	22	41	47	57	9	18	20	44	43	24	46	60
1	42	40	26	27	37	23	64	55	20	28	33	29	40	45	10
63	50	29	35	34	32	15	2	54	44	38	31	35	26	21	11
62	49	33	31	30	36	16	3	12	43	39	30	34	27	22	53
4	17	28	38	39	25	48	61	13	24	25	36	32	37	41	52
5	18	20	44	43	24	46	60	51	46	48	16	15	23	47	14
59	9	55	54	12	13	51	7	7	60	61	3	2	64	57	6



# 各魔方陣のシミュレーション方法・結果(8)

## ◆ 同心魔方陣

- アルゴリズム

A1	B1
B2	A2

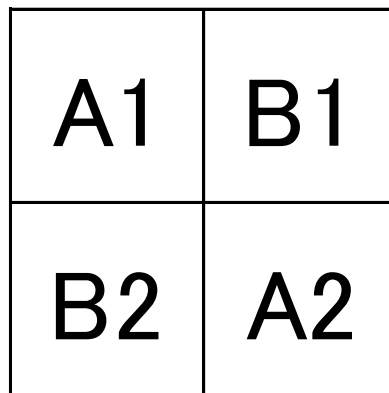
59	5	4	62	63	1	8	58	58	56	10	11	53	52	14	6
9	18	17	49	50	42	19	56	8	19	45	21	22	41	47	57
55	20	28	33	29	40	45	10	1	42	40	26	27	37	23	64
54	44	38	31	35	26	21	11	63	50	29	35	34	32	15	2
12	43	39	30	34	27	22	53	62	49	33	31	30	36	16	3
13	24	25	36	32	37	41	52	4	17	28	38	39	25	48	61
51	46	48	16	15	23	47	14	5	18	20	44	43	24	46	60
7	60	61	3	2	64	57	6	59	9	55	54	12	13	51	7
58	56	10	11	53	52	14	6	59	5	4	62	63	1	8	58
8	19	45	21	22	41	47	57	9	18	17	49	50	42	19	56
1	42	40	26	27	37	23	64	55	20	28	33	29	40	45	10
63	50	29	35	34	32	15	2	54	44	38	31	35	26	21	11
62	49	33	31	30	36	16	3	12	43	39	30	34	27	22	53
4	17	28	38	39	25	48	61	13	24	25	36	32	37	41	52
5	18	20	44	43	24	46	60	51	46	48	16	15	23	47	14
59	9	55	54	12	13	51	7	7	60	61	3	2	64	57	6

# 各魔方陣のシミュレーション方法・結果(9)

## ◆ 同心魔方陣

### • アルゴリズム

1. A1の1
2. A2の1
3. B1の1
4. B2の1
5. A1の2
- ⋮
255. B1の256
256. B2の256



59	5	4	62	63	1	8	58	58	56	10	11	53	52	14	6
9	18	17	49	50	42	19	56	8	19	45	21	22	41	47	57
55	20	28	33	29	40	45	10	1	42	40	26	27	37	23	64
54	44	38	31	35	26	21	11	63	50	29	35	34	32	15	2
12	43	39	30	34	27	22	53	62	49	33	31	30	36	16	3
13	24	25	36	32	37	41	52	4	17	28	38	39	25	48	61
51	46	48	16	15	23	47	14	5	18	20	44	43	24	46	60
7	60	61	3	2	64	57	6	59	9	55	54	12	13	51	7
58	56	10	11	53	52	14	6	59	5	4	62	63	1	8	58
8	19	45	21	22	41	47	57	9	18	17	49	50	42	19	56
1	42	40	26	27	37	23	64	55	20	28	33	29	40	45	10
63	50	29	35	34	32	15	2	54	44	38	31	35	26	21	11
62	49	33	31	30	36	16	3	12	43	39	30	34	27	22	53
4	17	28	38	39	25	48	61	13	24	25	36	32	37	41	52
5	18	20	44	43	24	46	60	51	46	48	16	15	23	47	14
59	9	55	54	12	13	51	7	7	60	61	3	2	64	57	6

中央と隅を取りつつ、擬似ランダムなスイッチングを再現

# 各魔方陣のシミュレーション方法・結果(10)

## ◆ 同心魔方陣

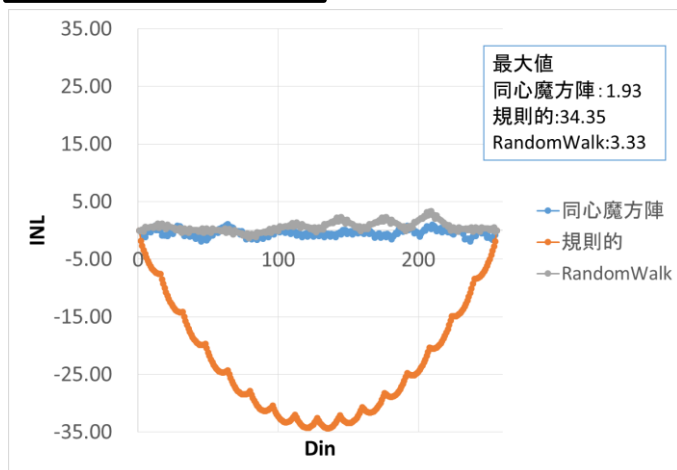
✓ Linear Error

$$\varepsilon_l(x, y) = g_l * \cos \theta * x + g_l * \sin \theta * y$$

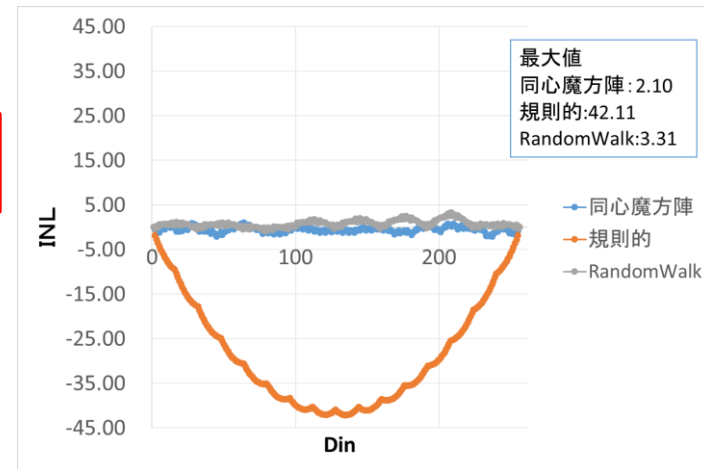
$$\theta = 30^\circ, 45^\circ, 60^\circ$$

$$g_l = 1$$

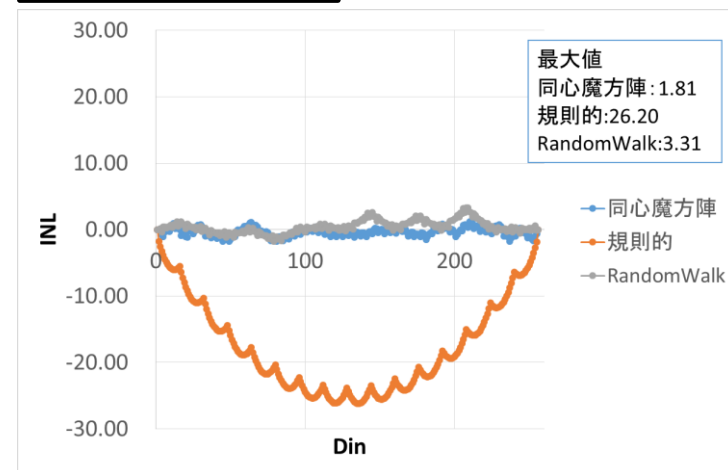
•  $\theta = 45^\circ$



•  $\theta = 30^\circ$



•  $\theta = 60^\circ$

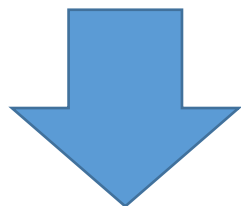


# 各魔方陣のシミュレーション方法・結果(11)

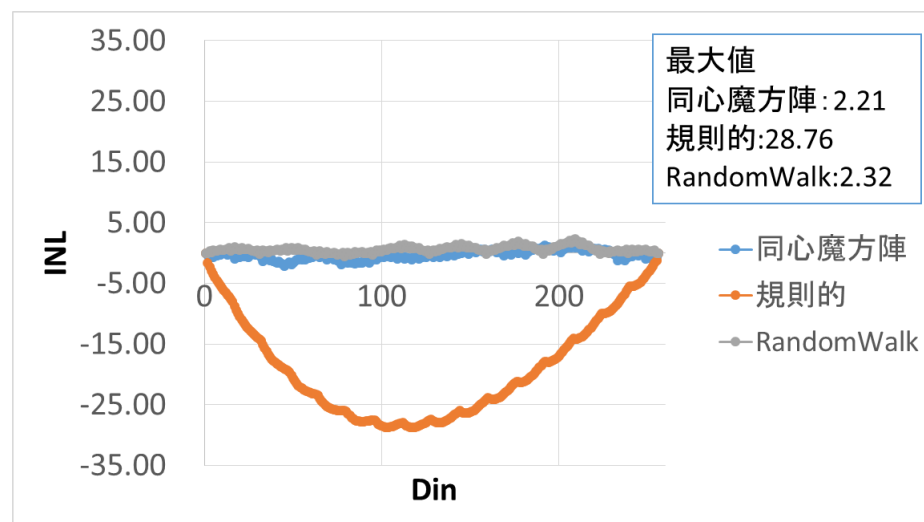
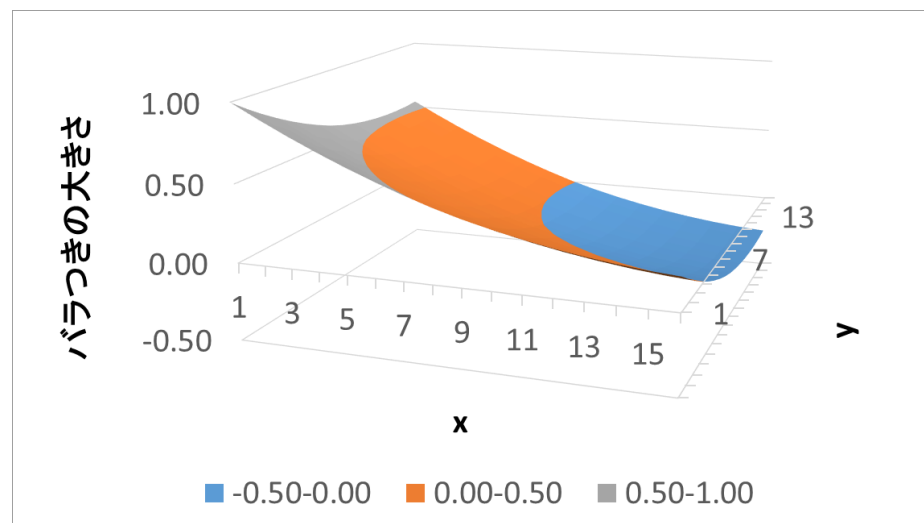
## ◆ 同心魔方陣

✓ Joint Error

一次 > 二次 の場合

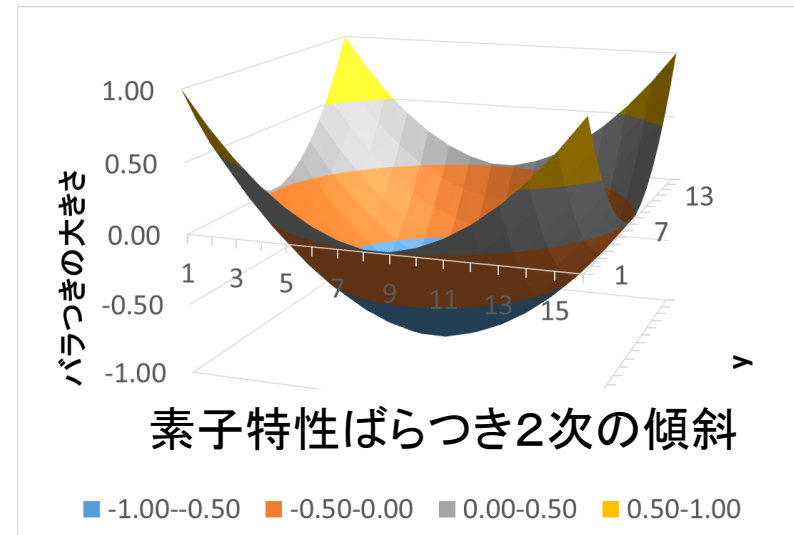
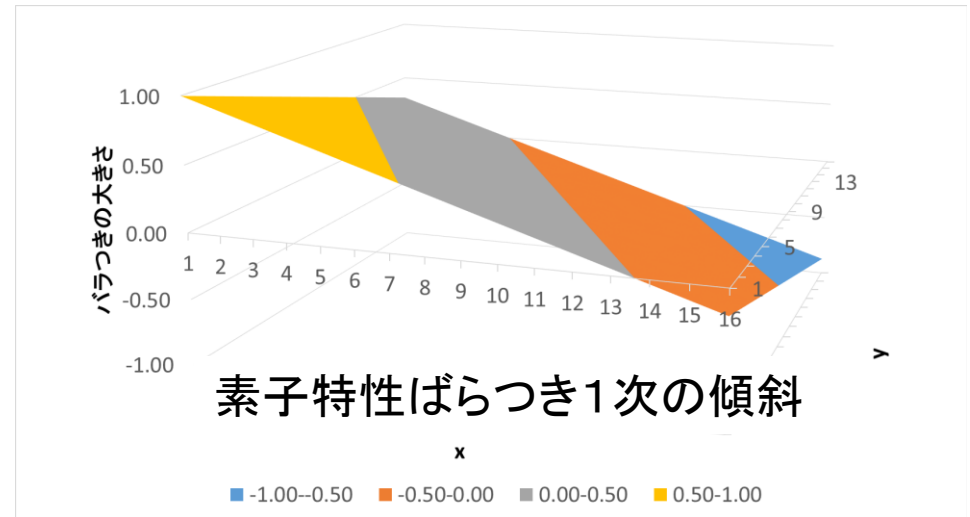


魔方陣の方が適している



# 魔方陣DA変換器レイアウト まとめ

- 魔方陣レイアウトは素子ばらつき1次の傾斜キャンセルにより有効
- 他の2次元技術(画像技術等)にも展開

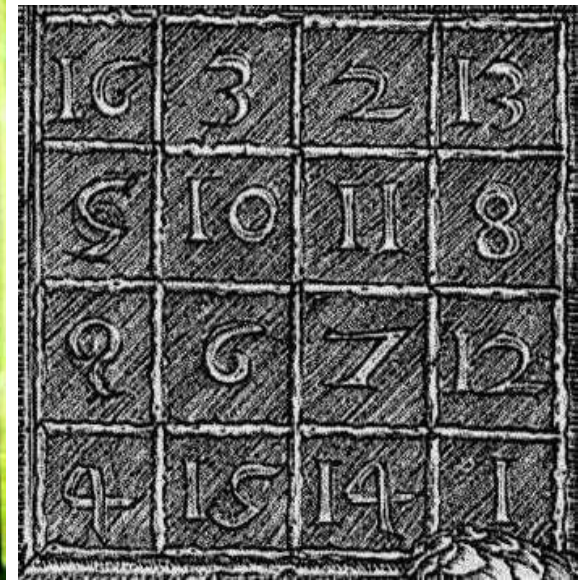
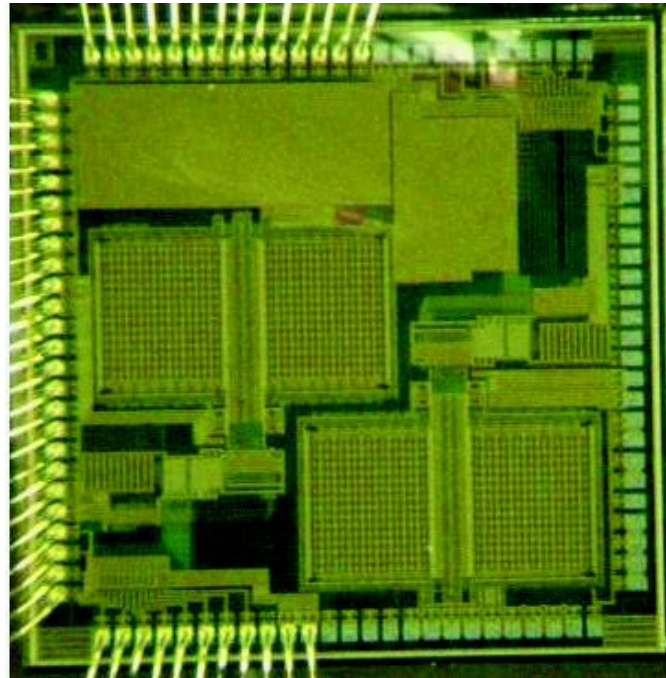
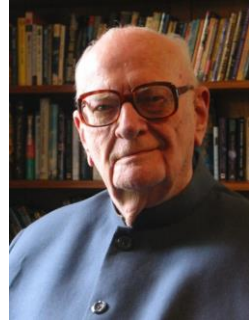




# まとめ

Any sufficiently advanced technology is indistinguishable from **magic**.

(科学小説作家 Arthur C. Clarke の第3法則)



Magic (手品) と Magic Square (魔方陣) をかけています

# 剰余系(孫子算経)を用いた 時間デジタル変換回路

李从兵 (群馬大学)      加藤健太郎 (鶴岡高専)  
王俊善      小林春夫 (群馬大学)

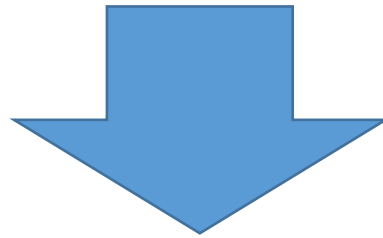


# 中国の剰余定理

- 中国の算術書『孫子算経』

「3で割ると2余り、5で割ると3余り、  
7で割ると2余る数は何か」

一般化



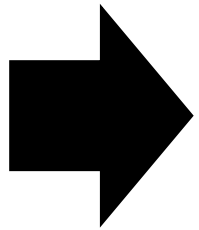
中国の剰余定理

答え 23



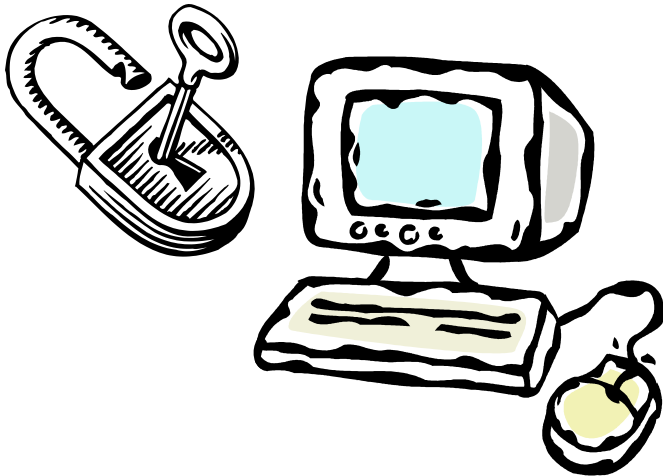
孫子算経

- ✓ 江戸時代、「百五減算」として伝来
- ✓ 現在、情報セキュリティの暗号化に応用



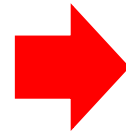
古典数学によるイノベーション

集積回路に応用

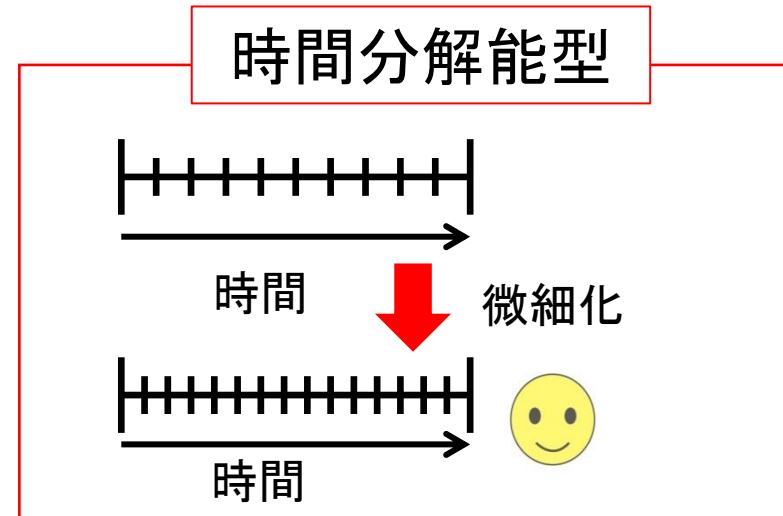
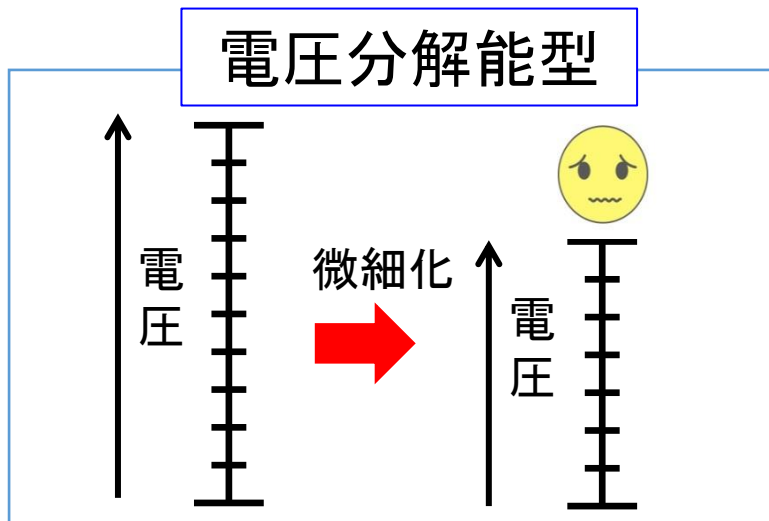


関孝和

微細化CMOS LSI



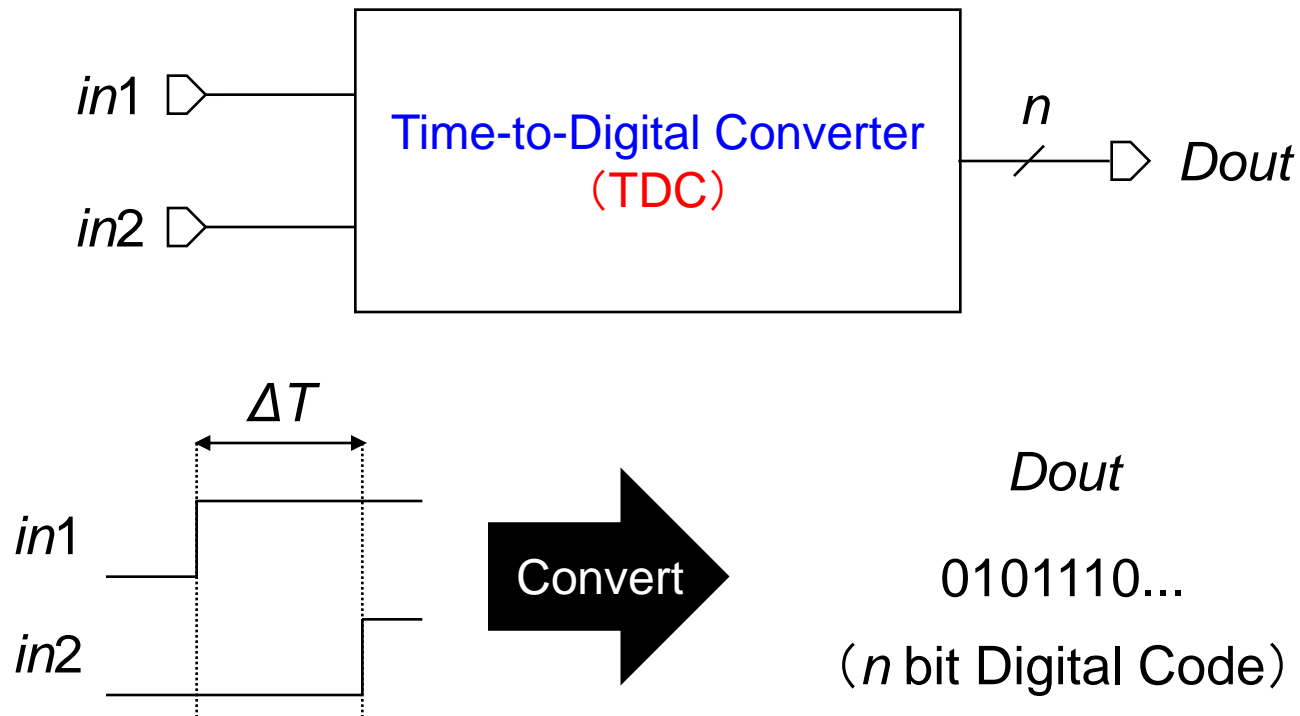
電源電圧の低下  
動作スイッチングスピードの向上



TDC (Time-to-Digital Converter) は2つのデジタル信号の時間差をデジタル値に変換



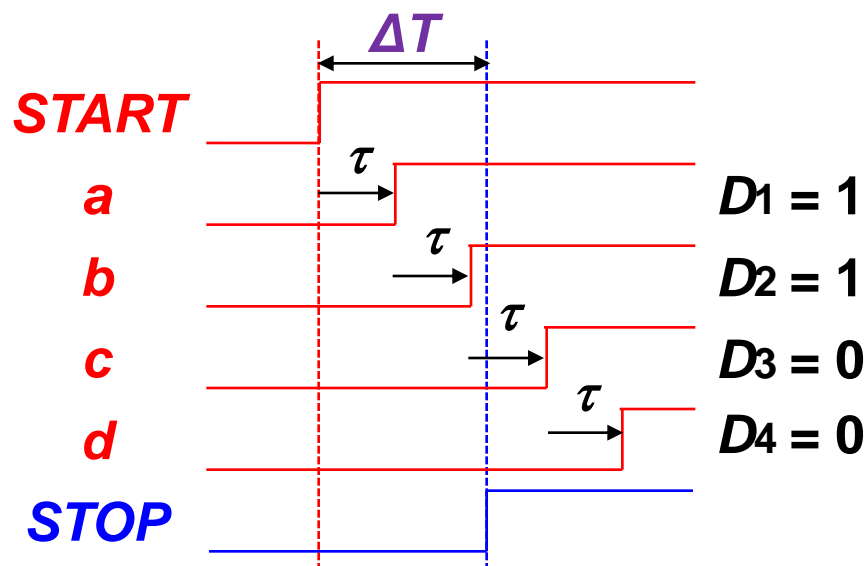
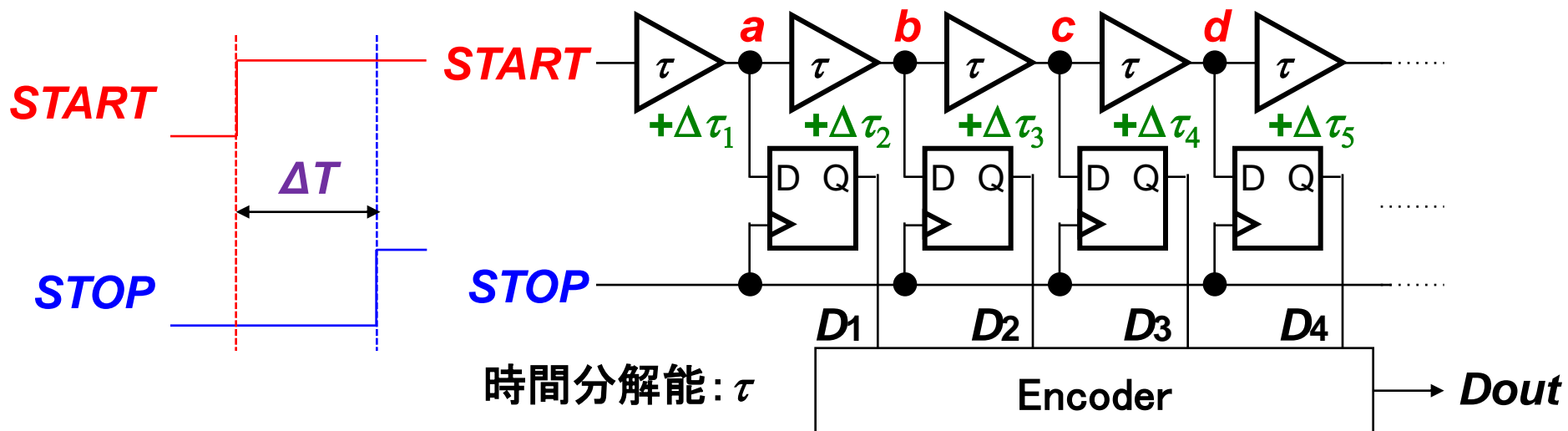
微細化CMOS LSIにおいて、TDCは時間領域アナログ回路のカギとなる  
(センサ回路, All-Digital PLL, ADC, 変調回路等)



2つのデジタル信号間の時間差  $\Delta T$  をデジタル値に変換

出力のデジタル値より  $\Delta T$  を測定可能

# フラッシュ型 TDCの構成と動作



- $\Delta T$  の大きさに比例したデジタル値  $D_{out}$  を出力

- 時間分解能  $\tau$

高エネルギー加速器研究機構  
素粒子原子核研究所  
新井康夫氏による発明

# フラッシュ型TDCの回路規模の問題

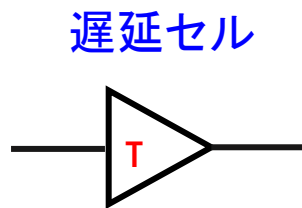
START とSTOP の立ち上がりエッジ間の時間差

測定範囲  $0 < \Delta T < N T$   $\Delta T$

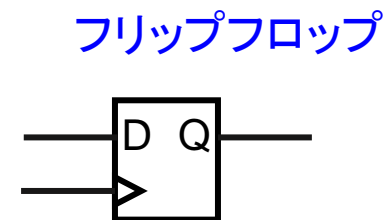
時間分解能  $T$

$N = 1001$  (千一) のとき

フラッシュ型TDC では大きな回路規模、大きな消費電力



1001個



1001個

**提案する剰余系TDC**  $1001 = 7 \times 11 \times 13$   
 同じ測定範囲、時間分解能で  $7 + 11 + 13 = 31$  個の  
 遅延セル、フリップフロップで実現できる

千一個から三十一個へ !!



# 研究の目的

## 時間測定回路TDC

- LSIテストシステムのキーコンポーネント
- 時間信号であることを利用

 “剰余”が容易に得られる

- 剰余系を利用

フラッシュ型TDCに比べ、  
同等性能、小回路規模・低消費電力TDCが  
実現できる可能性あり



剰余系TDC回路を検討

# 剰余系の例

基数 2, 3, 5 互いに素

$$N=2 \times 3 \times 5 = 30$$

0から $N-1(=29)$ までの整数の一つを  $k$

$$a: k \text{ を } 2 \text{ で割った余り} \quad a = \text{mod}_2(k)$$

$$b: k \text{ を } 3 \text{ で割った余り} \quad b = \text{mod}_3(k)$$

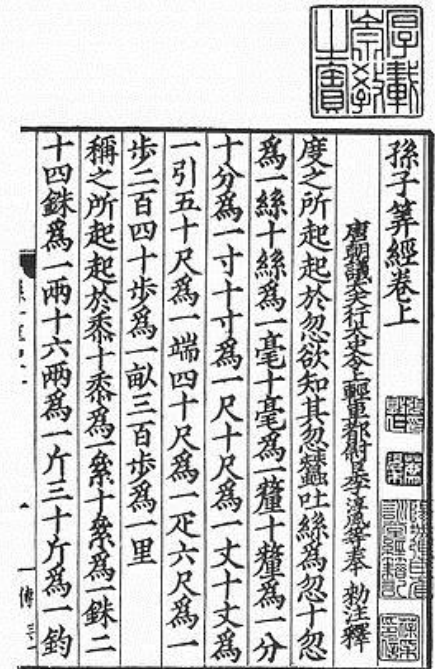
$$c: k \text{ を } 5 \text{ で割った余り} \quad c = \text{mod}_5(k)$$

$k$  と  $(a, b, c)$  の組は1対1に対応する。

$k$  を  $(a, b, c)$  で表現  剰余表現

中国人の剰余定理 (Chinese Remainder Theorem)

$(a, b, c)$  から  $k$  を求めるアルゴリズム



# 剰余定理の例

基数 2, 3, 5 互いに素

$$N=2 \times 3 \times 5 = 30$$

0から $N-1(=29)$ までの整数の一つを  $k$

$a$ :  $k$ を2で割った余り  $a = \text{mod}2(k)$

$b$ :  $k$ を3で割った余り  $b = \text{mod}3(k)$

$c$ :  $k$ を5で割った余り  $c = \text{mod}5(k)$

$k$ と $(a, b, c)$ の組は1対1に対応する。

$k$ を $(a, b, c)$ で表現  $\rightarrow$  剰余表現

自然数  $k$  と剰余表現  $(m_1, m_2, m_3)$  は 1対1 対応

$m_1$	$m_2$	$m_3$	$k$
0	0	0	0
1	1	1	1
0	2	2	2
1	0	3	3
0	1	4	4
1	2	0	5
0	0	1	6
1	1	2	7
0	2	3	8
1	0	4	9
0	1	0	10
1	2	1	11
0	0	2	12
1	1	3	13
0	2	4	14

$m_1$	$m_2$	$m_3$	$k$
1	0	0	15
0	1	1	16
1	2	2	17
0	0	3	18
1	1	4	19
0	2	0	20
1	0	1	21
0	1	2	22
1	2	3	23
0	0	4	24
1	1	0	25
0	2	1	26
1	0	2	27
0	1	3	28
1	2	4	29

剰余定理 (Chinese Remainder Theorem)

$(a, b, c)$  から  $k$  を求めるアルゴリズム

剰余定理は、

この問題を他の整数についても適用できるように一般化したもの。

# 剰余DCの原理

TDC 回路は信号が時間であることを利用すると“剰余”が容易に得られる。

三つのリング発振回路(遅延 $m_1T$ ,  $m_2T$ ,  $m_3T$ )を利用し、発振状態から経過時間 $T$ の測定を行うことが可能で。剰余定理に基づいて、(a, b, c)から $k$ を求め、経過時間 $T = k \times T$  を得る。

例えば、三つのリング発振回路(遅延 $2T$ ,  $3T$ ,  $5T$ )を利用し、発振している状態から経過時間 $T$ の測定を行う。

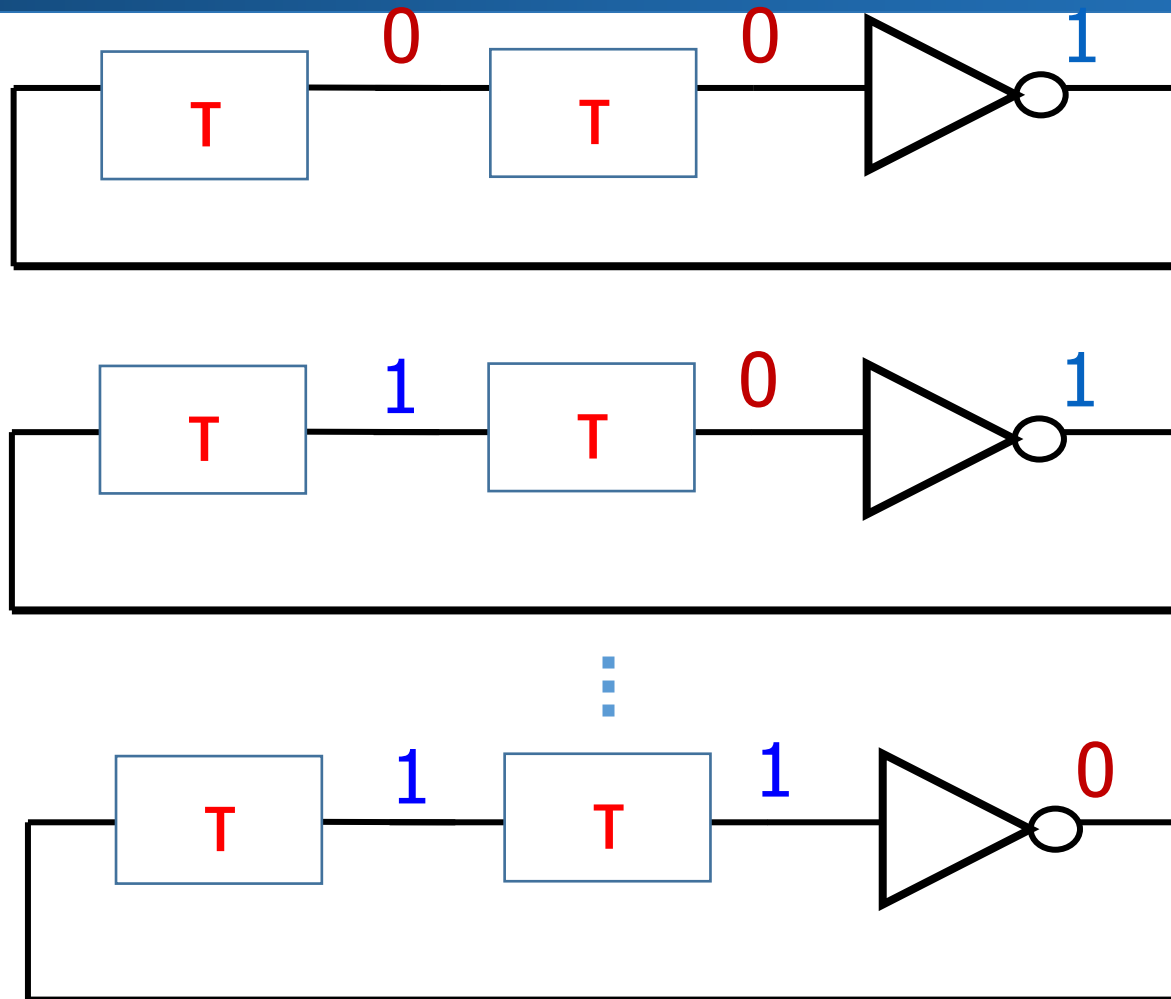
$T$  を $2T$ で割った余りは $a$

$T$  を $3T$ で割った余りは $b$

$T$  を $5T$ で割った余りは $c$

⇒剰余定理で  $T = k^*T$

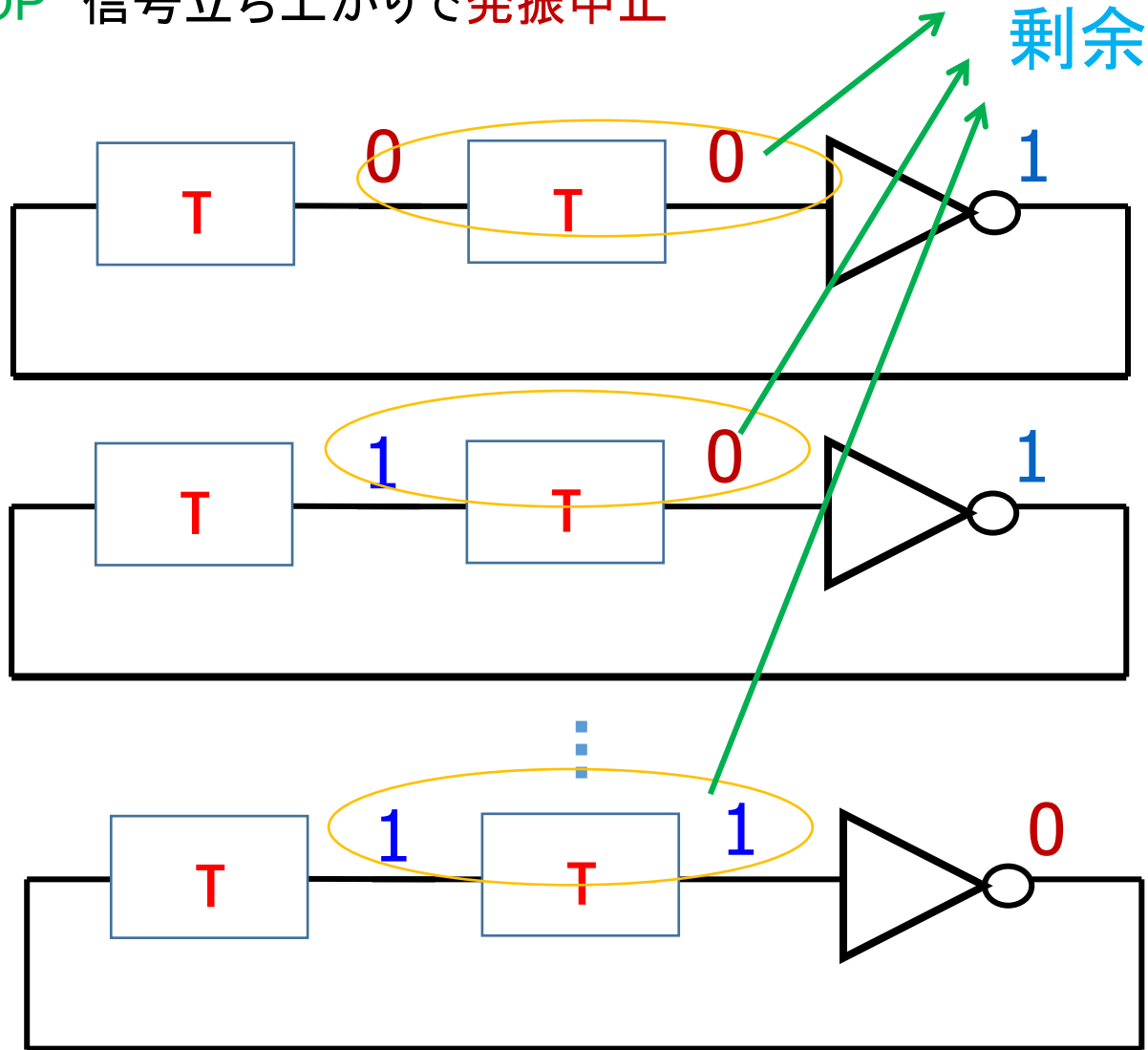
# リング発振回路で剰余が容易に得られる



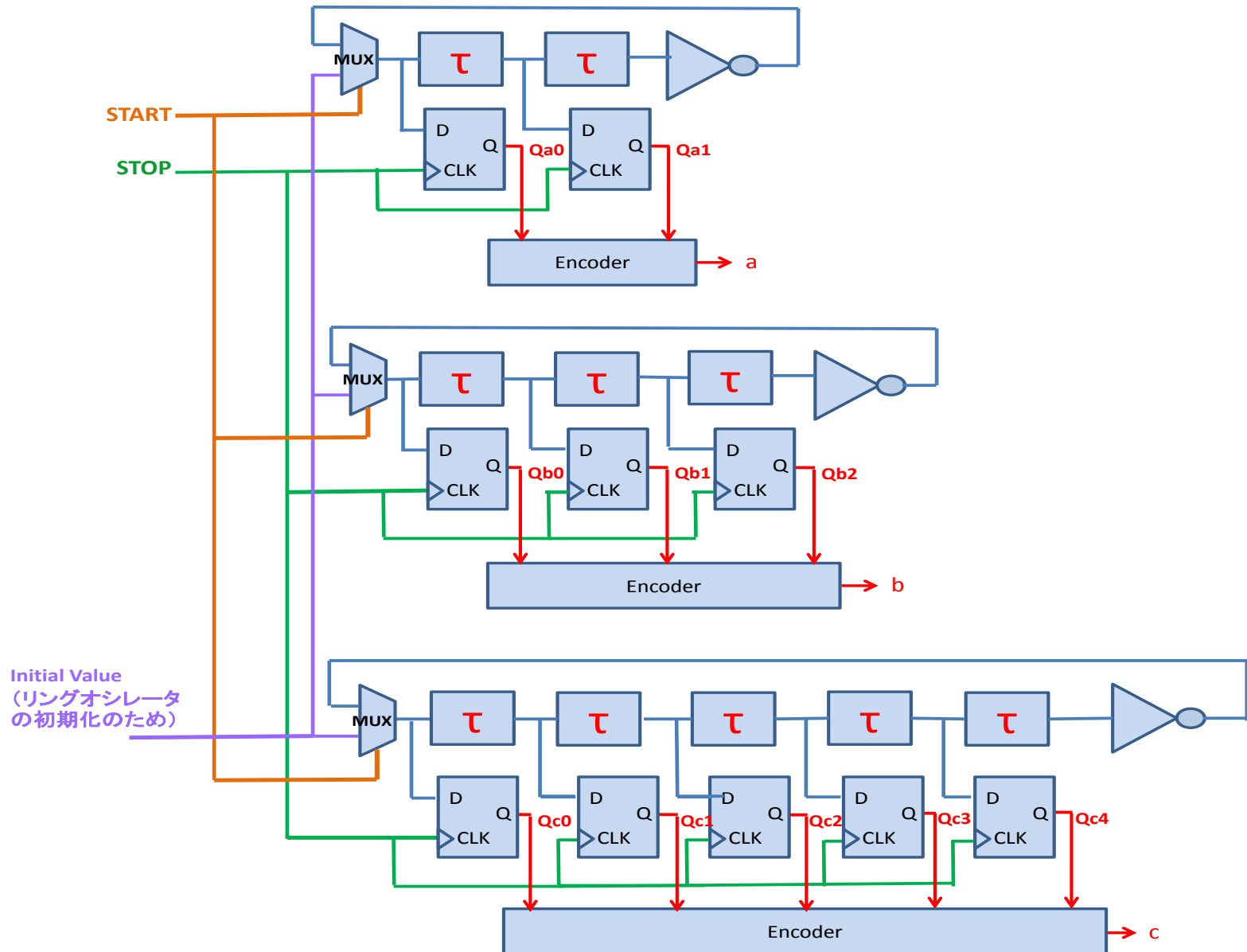
**考察** TDCでは取り扱う入力信号が時間信号なのでリング発振回路構成により剰余が容易に得られる。電圧信号を入力とするADCでは剰余を得るのは簡単ではない。

# リング発振回路で剰余を得る

- **START** 信号立ち上がりで発振開始
- **STOP** 信号立ち上がりで発振中止



## 提案する剰余系TDCの回路図







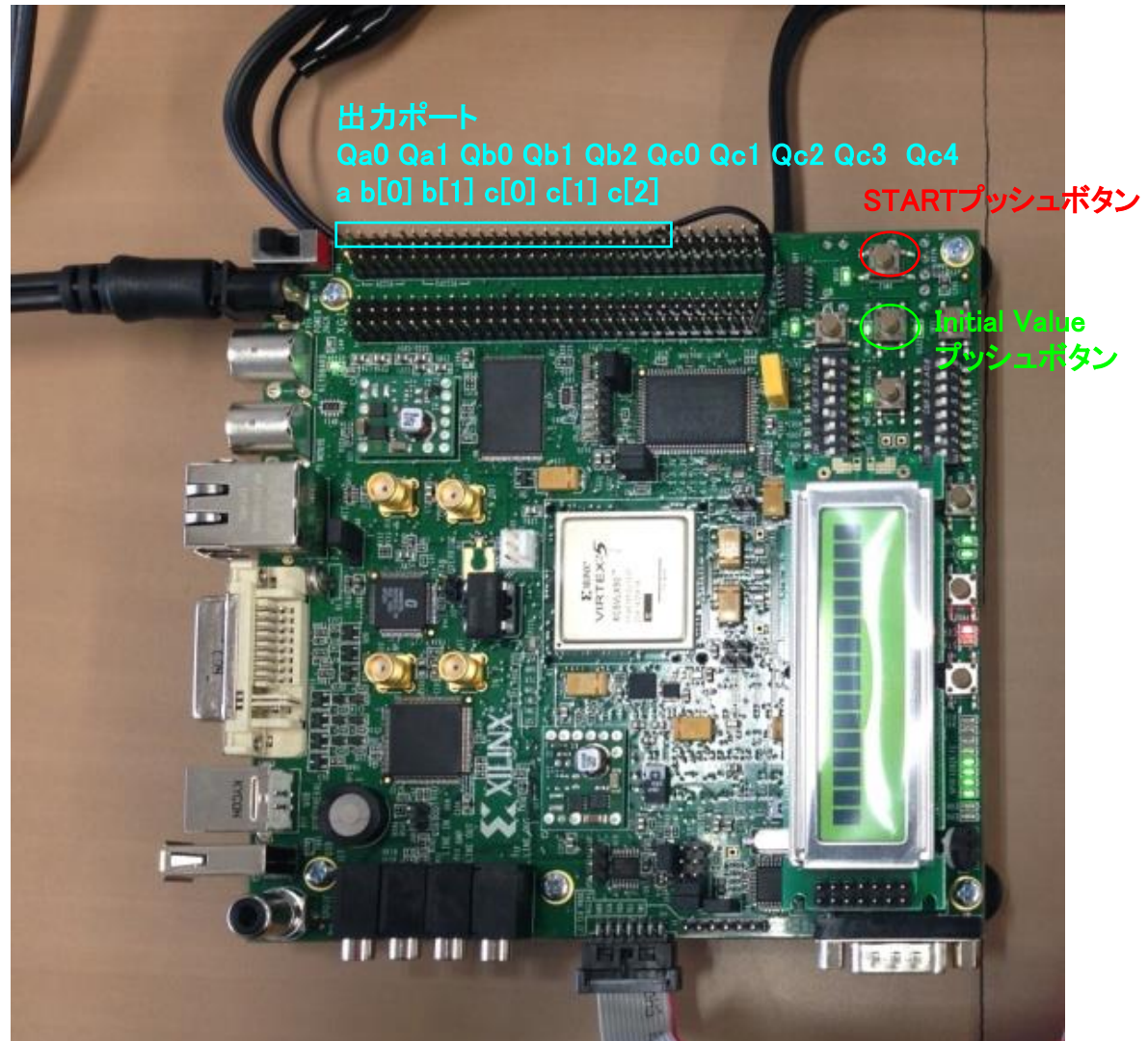
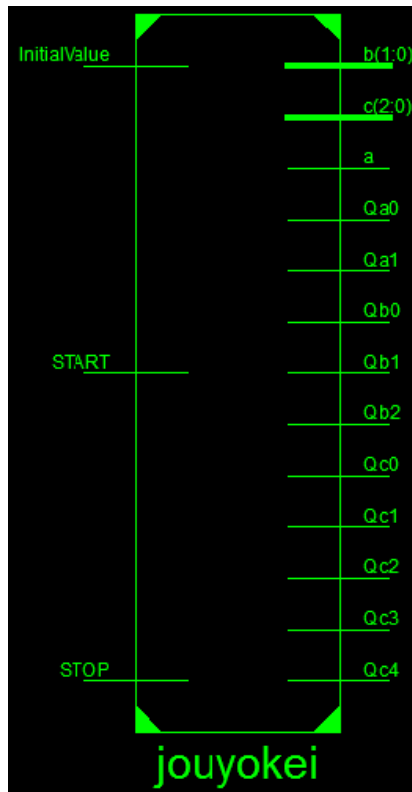
# FPGA実装

STOPポートの入力: 100MHz FPGA クロック

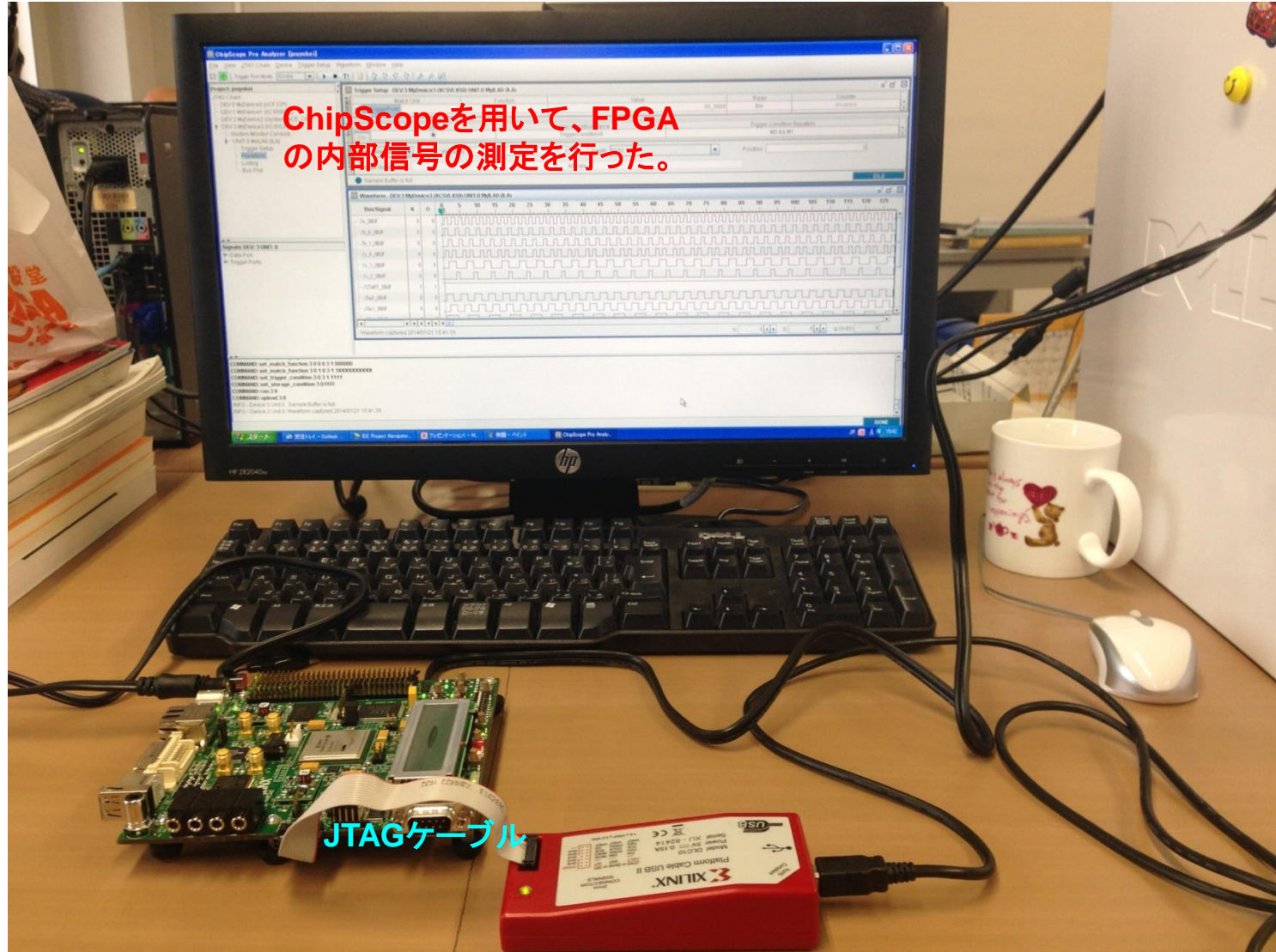
Buffer\_CLKポートの入力: 33MHz FPGA クロック(バッファの遅延  $\tau = 30.30\text{ns}$ )

入力ポート

出力ポート

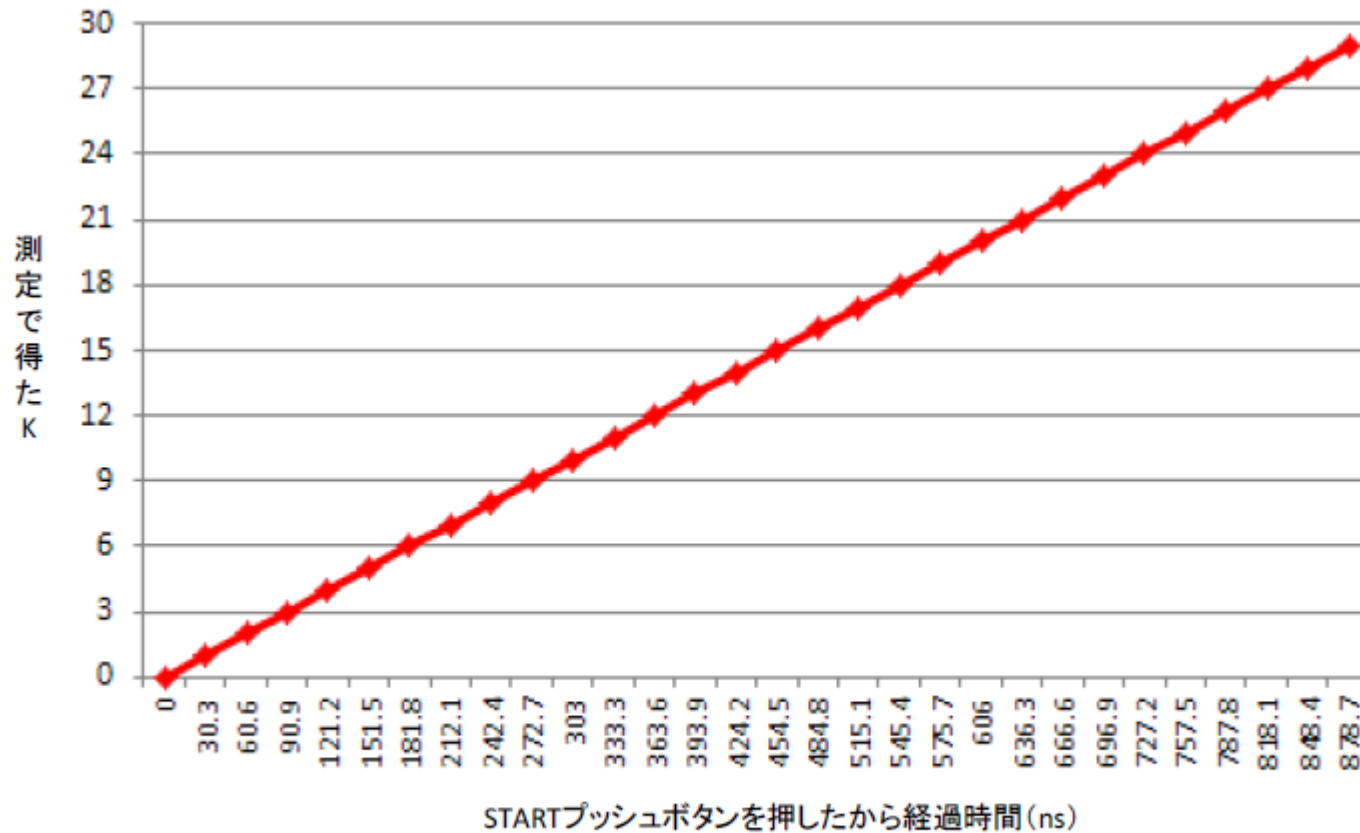


# FPGA実装



# FPGA実装 剰余系TDCの評価

剰余系TDC回路はFPGAで実現できることが示された。



経過時間 VS. 測定で得た k

# まとめと今後の課題

- 剰余系TDC回路は時間測定可を確認。  
FPGA 実現を示した。
- バッファの個数とリング発振回路の個数を増減し  
他の剰余数系にも適用可。
- 剰余系を利用したTDC 回路は、  
回路面積、消費電力が大幅低減可。

# 発表内容

- はじめに
- フィボナッチ数列と黄金比  
逐次比較近似AD変換器 冗長アルゴリズム
- 魔方陣  
セグメント型DA変換器 レイアウト
- 剰余系(孫子算経)  
時間デジタル変換回路
- まとめ

# 整数論の工学応用



Carolus Fridericus Gauss  
(独: 1777-1855)

「整数論は数学の女王である。」

カール・フリードリヒ・ガウス

過去の整数論

身近にあるが、謎が多く美しい。

他分野へ貢献しない孤高の学問。

現在の整数論

情報通信処理に応用(暗号化・符号論)

⇒ デジタル信号との相性良し

**AD/DA変換器, TDC回路への整数論応用は未知の世界  
今後大きな発見が待っている可能性**

# 参考文献

## フィボナッチ数列、黄金比 / 白銀比

- [1] Mario Livio (著)、齊藤 隆央 (翻訳): 黄金比はすべてを美しくするか？  
最も謎めいた「比率」をめぐる数学物語、早川書房、2012年1月.
- [2] ルフレッド・S・ポザマンティエ、イングマル・レーマン (著)、松浦俊輔 (翻訳):  
不思議な数列フィボナッチの秘密、日経BP社、2010年8月.
- [3] 桜井 進: “雪月花の数学”、祥伝社、2010年6月.
- [4] Keith Ball (原著)、佐藤 かおり (翻訳): フィボナッチのうさぎ—数学探険旅行、青土社、  
2006年12月.
- [5] 中村 滋 : フィボナッチ数の小宇宙、日本評論社、2002年9月.

## 冗長逐次比較AD/DA変換器

- [6] F. Kuttner, “A 1.2V 10b 20MSample/s non-binary successive approximation ADC in 0.13 $\mu$ m CMOS,” Tech. Digest of ISSCC, (Feb. 2007).
- [7] T. Ogawa, H. Kobayashi, Y. Takahashi, N. Takai, M. Hotta, H. San, T. Matsuura, A. Abe, K. Yagi, T. Mori : “SAR ADC Algorithm with Redundancy and Digital Error Correction”,  
EICE Trans. Fundamentals, vol.E93-A, no.2, (Feb. 2010).
- [8] 小川 智彦, 松浦 達治, 小林 春夫, 高井 伸和, 堀田 正生, 傘 昊, 阿部 彰, 八木 勝義, 森 俊彦,  
“逐次比較近似ADC コンパレータ・オフセット影響の冗長アルゴリズムによるデジタル補正技術,”  
電子情報通信学会誌 和文誌C, Vol.J94-C, No.3 (2011年3月)

# 参考文献

- [9] 小林佑太郎, 香積正基, 楊志翔, 小林春夫: ” ADC/DAC のフィボナッチ数列を用いた冗長性設計の検討”, 電気学会 電子回路研究会, ECT-13-087, 奈良 (2013年10月).
- [10] 楊志翔, 小林佑太郎, シャイフルニザムビンモーヤ, 小林春夫, 「フィボナッチ数列を用いたDA変換回路アーキテクチャ」 電気学会 電子回路研究会, ECT-13-087, 奈良 (2013年10月).
- [11] 小林佑太郎, 小林春夫: “黄金分割法を用いたSAR ADC冗長設計”, 第4回 電気学会 東京支部 栃木・群馬支所 合同研究発表会, ETT-14-28, ETG-14-28 (2014年3月).
- [12] 小林佑太郎, 小林春夫: ” 逐次比較近似ADCの整数論に基づく冗長アルゴリズム設計”, 電気学会 電子回路研究会, ECT-14-059, 島根 (2014年7月).
- [13] Y. Kobayashi, H. Kobayashi, “SAR ADC Algorithm with Redundancy Based on Fibonacci Sequence”, The 3rd Solid State Systems Symposium-VLSIs and Semiconductor Related Technologies & The 17th International Conference on Analog VLSI Circuits, Ho Chi Minh City, Vietnam (Oct. 2014)

## 魔方陣、魔方陣を用いたDA変換器レイアウト

- [14] 大森清美, ”魔方陣の世界”, 日本評論社, 2013年8月10日
- [15] 東野将史, 小林春夫, ”セグメント型DA変換器の魔方陣レイアウト技術による線形性向上”, 電気学会 電子回路研究会, 2014年10月9, 10日

## 剰余系時間デジタイザ回路

- [16] C. Li, K. Katoh, H. Kobayashi, J. Wang, S. Wu, S. N. Mohyar, “Time-to-Digital Converter Architecture with Residue Arithmetic and its FPGA Implementation”, 11th International SoC Design Conference, Jeju, Korea (Nov. 2014).



# (付録1) 日本の美しさ“白銀比”

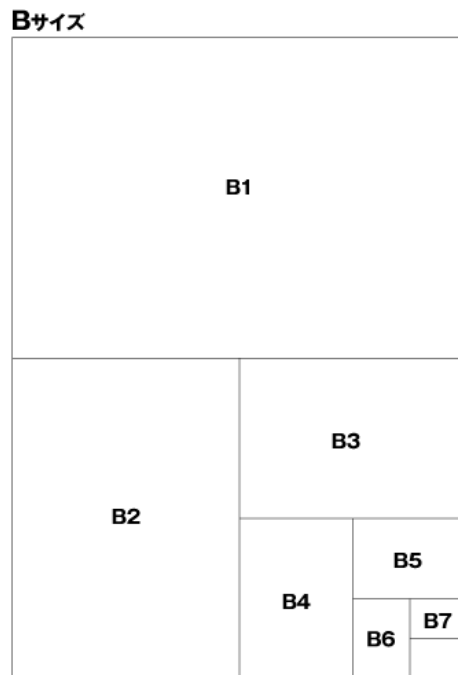
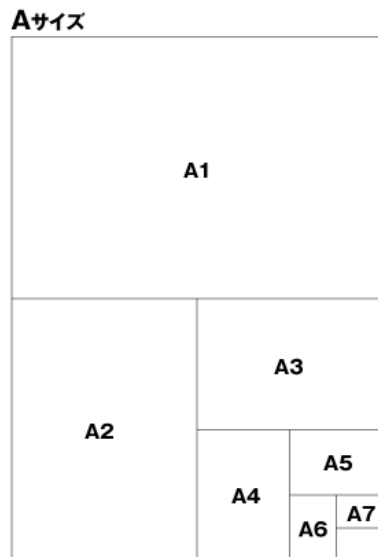
日本人が思う美しい比率

$$\frac{\sqrt{2}}{1} = 1.414 = \text{白銀比}$$

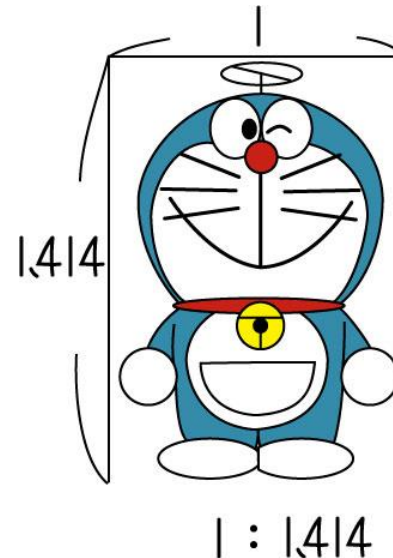
コピー用紙  
A版、B版とも 縦横比率が白銀比

ドラえもん、アンパンマン、トトロ、  
クレヨンしんちゃん が白銀比

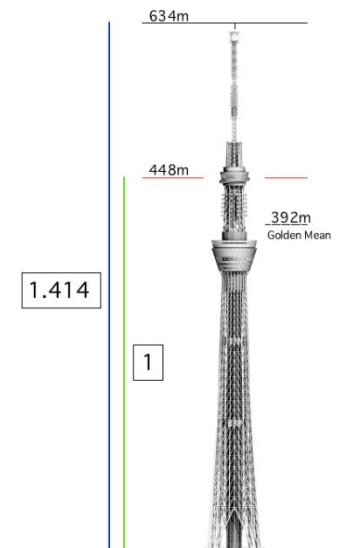
コピー用紙



キャラクター



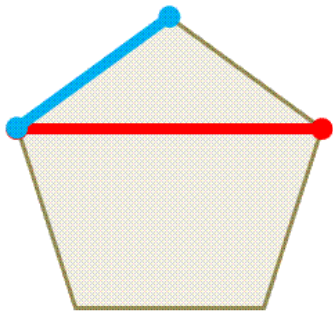
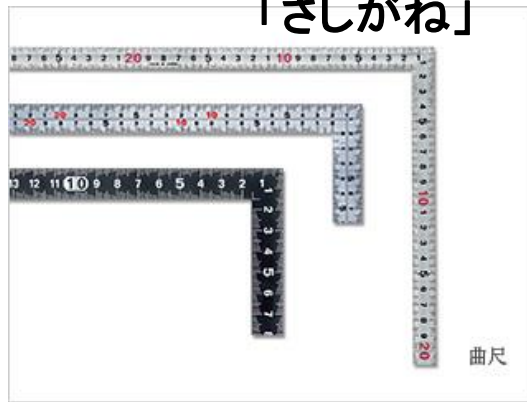
スカイツリー



# (付録1) 日本の美しさ“白銀比” (続き)

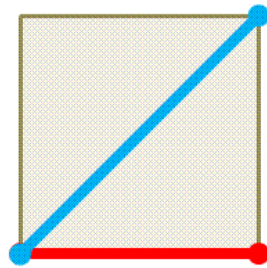
日本人は風呂敷、畳、法隆寺等 正方形を好む。

日本古来の工具  
「さしがね」



黄金比

1 : 0.618



白銀比

1 : 1.414

法隆寺

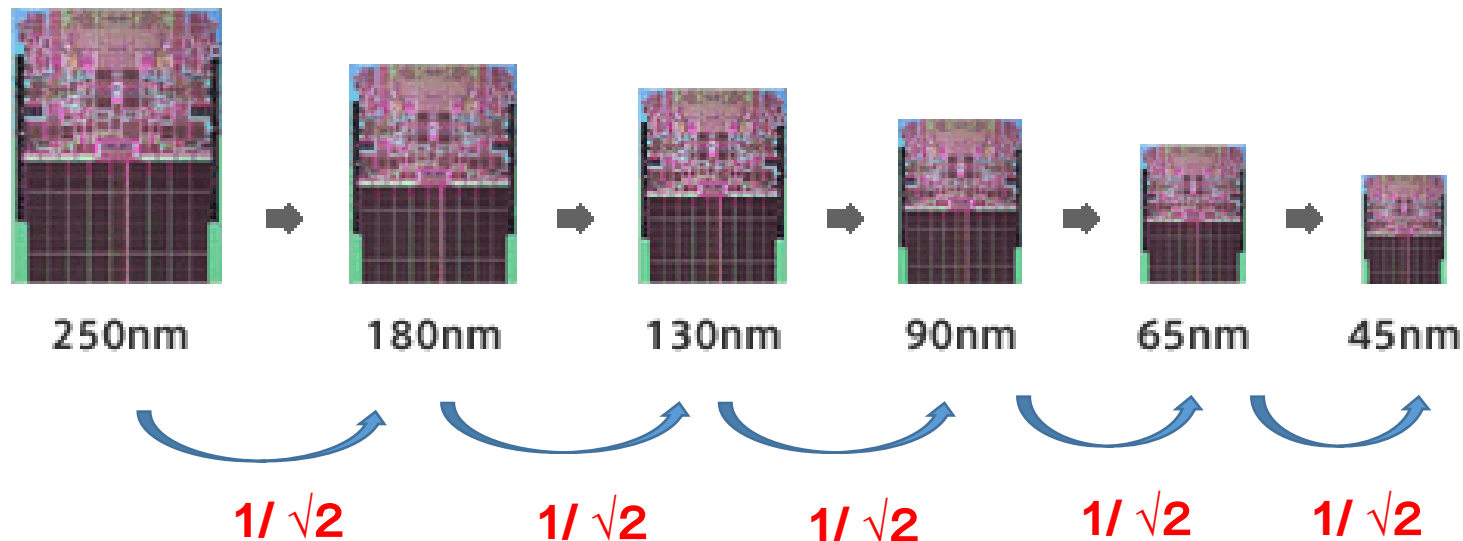


正方形： 丸い材木から  
最も無駄を出さない切り方。

無駄を好まない島国の心。

# (付録2) 集積回路と“白銀比”

## LSIプロセス進展



黄金比は動的な比率 (無限の螺旋)

⇔ 白銀比は静的な比率 (合理化)

両者に発展の可能性が存在する。

# 講演終了後 聴衆の方より

「自然界が1.6進(黄金比 進)ならば、  
2進で計測するというのは  
無理があるかもしれないと思った。  
1.6進(黄金比 進)のほうが良い計測が  
できそうだというのはなるほどと思った。」