

ADC/DACのフィボナッチ数列を用いた冗長性設計の検討

◎小林 佑太郎 香積 正基

楊 志翔 小林 春夫

群馬大学

- 研究背景、目的
- 冗長性を持つSAR ADCについて
- フィボナッチ数列を用いた冗長性設計
- 理論シミュレーション結果
- 電流源DACへの応用
- まとめ、今後の課題

- 研究背景、目的
- 冗長性を持つSAR ADCについて
- フィボナッチ数列を用いた冗長性設計
- 理論シミュレーション結果
- 電流源DACへの応用
- まとめ、今後の課題

自動車のエレクトロニクス化によって
車載用マイコン技術産業が発展
マイコン用ADCへの要求が厳しい

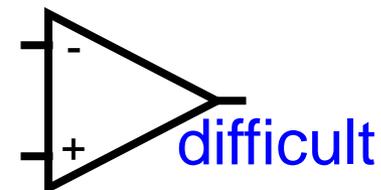
要求される項目は...

- 高分解能
- 高速
- 高精度
- 高信頼性
- 低コスト
- 低消費電力
- 小面積(高集積)



どの方式のADCを考えるか

微細化によって高性能オペアンプ設計が困難



ナノCMOSでの実現にもっとも適した
オペアンプを使わないADC



大部分が
デジタル回路構成

SAR ADC: 逐次比較方式AD変換器

高分解能・低消費電力・小型小チップ面積・中速

SAR ADC

冗長性(余裕)の利用

時間の冗長性 → デジタル誤差補正実現

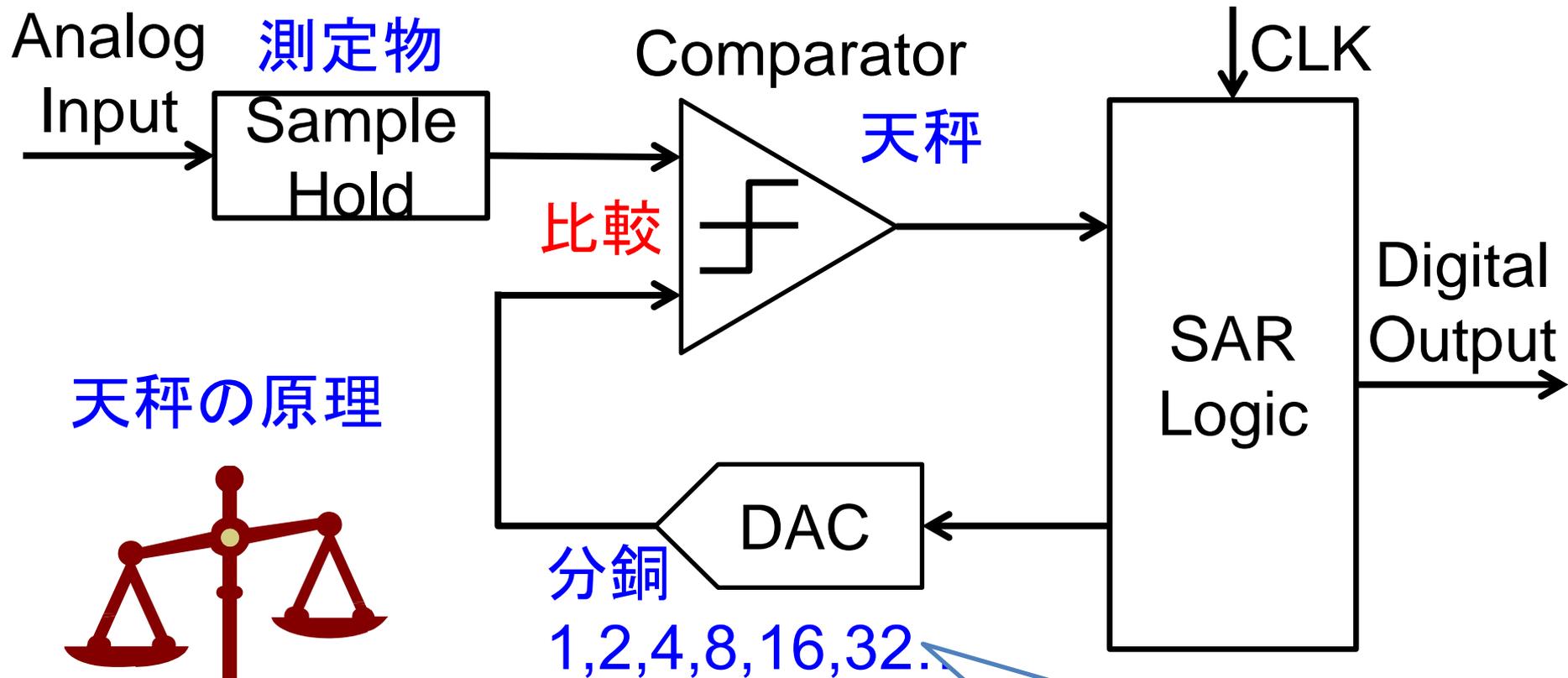
フィボナッチ数列で冗長性設計理論を検討
高信頼性化・高速化を目指す！

- 研究背景、目的
- 冗長性を持つSAR ADCについて
- フィボナッチ数列を用いた冗長性設計
- 理論シミュレーション結果
- 電流源DACへの応用
- まとめ、今後の課題

SAR ADCとは

ECT-13-087

アナログ入力と参照電圧を比較、結果に応じたデジタル出力



天秤の原理



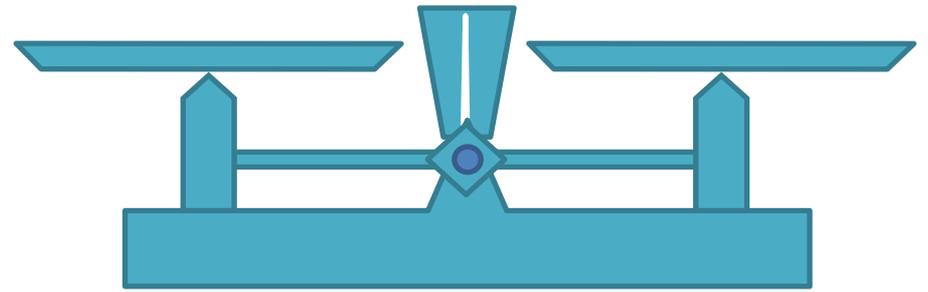
参照電圧の大きさは
二進重みで無駄がない

4bit二進探索SAR ADC動作原理

ECT-13-087

ステップ数		1	2	3	4
比較電圧の重み		8	4	2	1
レベル	15				
	14				
	13		0		
	12		0		
	11				
	10				1
	9			1	1
	8			1	
	7	1			
	6	1			
	5				
	4				
	3				
	2				
	1				
	0				

アナログ入力: **11.4**



比較電圧(二進重み)



デジタル出力 **1 0 1 1**

8 + 0 + 2 + 1 → **11**₉

4bit二進探索SAR ADC動作

ECT-13-087

コンパレータの誤動作を考えると...

4bit分解能
4step
2進重み
(8, 4, 2, 1)
アナログ入力
 $V_{in}=11.4V$

ステップ数		1	2	3	4
比較電圧の重み		8	4	2	1
レベル	15				
	14				
	13				
	12				
	11				
	10				
	9				
	8				
	7				
	6				
	5				
	4				
	3				
	2				
	1				
	0		0	1	1

コンパレータ
判定間違い

$$\begin{aligned} D_{out} &= (0111)_2 \\ &= (7)_{10} \end{aligned}$$

一回の判定誤り
誤りを出力
信頼性低い

改善

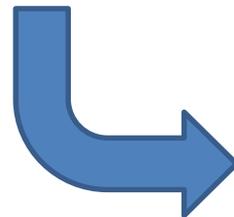
冗長: 余分や余裕のこと

↓ SAR ADCに適用

時間の冗長性を利用
判定ステップ数を増加

↓ 4step ⇒ 5step など

一つの数の表現方法増加



誤り耐性向上
結果的な速度上昇

二進重み 1,2,4,8



非二進重み 1,2,3,5,8

冗長探索SAR ADC動作

ECT-13-087

4bit分解能

4bit5step

非2進重み

(8, 5, 3, 2, 1)

アナログ入力

$V_{in}=11.4V$

ステップ数		1	2	3	4	5
比較電圧の重み		8	5	3	2	1
レベル	15					
	14					
	13					
	12					
	11					
	10					
	9					
	8					
	7					
	6					
	5					
	4					
	3					
	2					
	1					
	0					

$$\begin{aligned} \text{Dout} &= 10100 \\ &= 8+3 \\ &\Rightarrow (11)_{10} \end{aligned}$$

冗長探索SAR ADC動作

ECT-13-087

4bit分解能

4bit5step

非2進重み

(8, 5, 3, 2, 1)

アナログ入力

$V_{in}=11.4V$

ステップ数		1	2	3	4	5	
比較電圧の重み		8	5	3	2	1	
レベル	15						
	14						
	13						
	12						
	11	Red line					Blue line
	10				Blue line		
	9						
	8	Red line		Blue line			
	7						
	6						
	5		Blue line				
	4						
	3						
	2						
	1		0	1	1	1	1
	0						

Dout
= 01111
= 5+3+2+1
⇒ (11)₁₀

コンパレータ
判定間違い

誤りを補正
信頼性向上

なぜ誤りを補正することができるのか？

4bit分解能(16レベル)において

比較	二進探索4bit4step 参照電圧重み(8, 4, 2, 1) (11) ₁₀ ⇒ 1011
	冗長探索4bit5step 参照電圧重み(8, 5, 3, 2, 1) (11) ₁₀ ⇒ 10100 , <u>1</u> 0011 , <u>0</u> 1111 <div style="text-align: center; border: 1px solid green; padding: 2px; margin-top: 5px;">重さ8を間違えてもよい</div>

デジタル出力11を表現する方法が複数ある



冗長性によって誤差補正を可能とする

冗長探索SAR ADCの速度

ECT-13-087

デジタル誤差補正によって
判定の間違いを許容



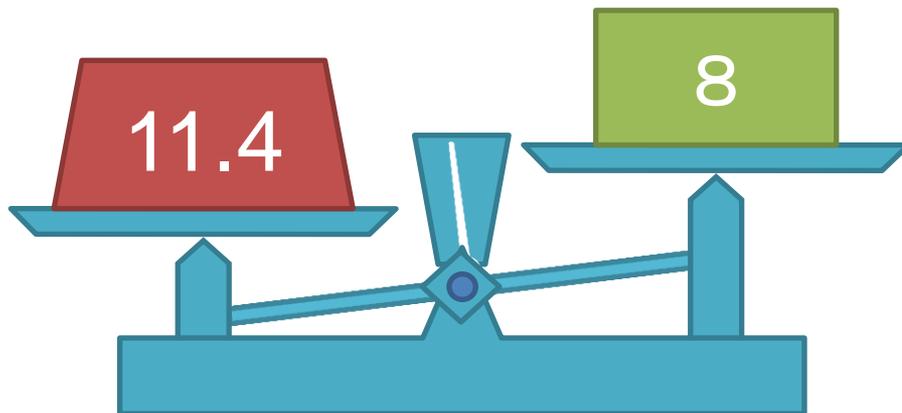
不完全整定許容
比較条件を緩和

4

デジタル
出力



1 1 . 4 と 1 2 の比較
判定が長時間



不完全に整定



後ろでデジタル補正

冗長探索SAR ADCの速度

ECT-13-087

デジタル誤差補正によって
判定の間違いを許容



不完全整定許容
比較条件を緩和

二進探索(完全整定)

5bit逐次比較方式ADC



AD変換時間

冗長探索(不完全整定)



不完全整定誤差補正

AD変換時間

結果的に**速度が上昇**する

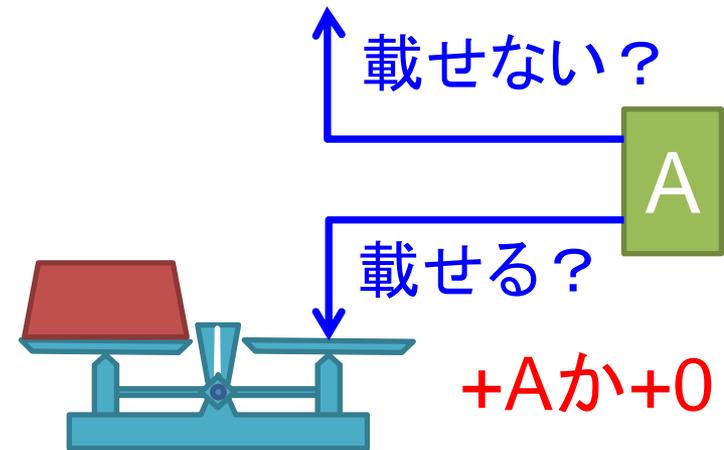
2つの補正方法

ECT-13-087

冗長性を利用すると補正方法が(天秤の考え方で) 2種類できる

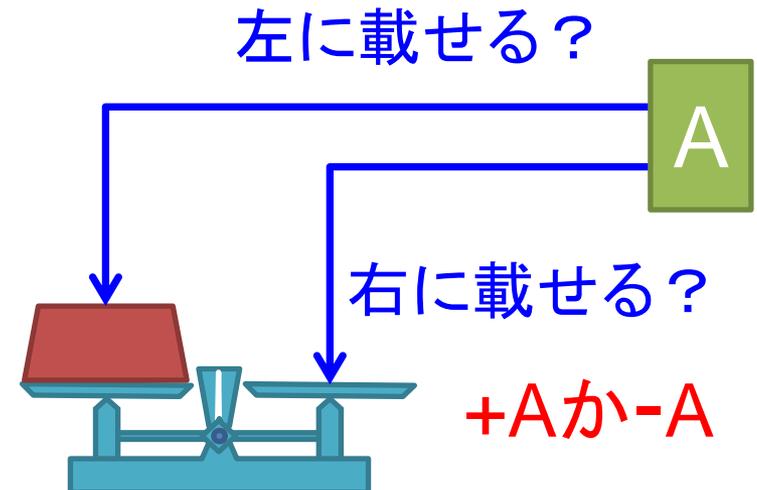
加算型

値を**加算**するか**加算**しないかを
決める方法



加減算型

値を**加算**するか**減算**するか
を決める方法



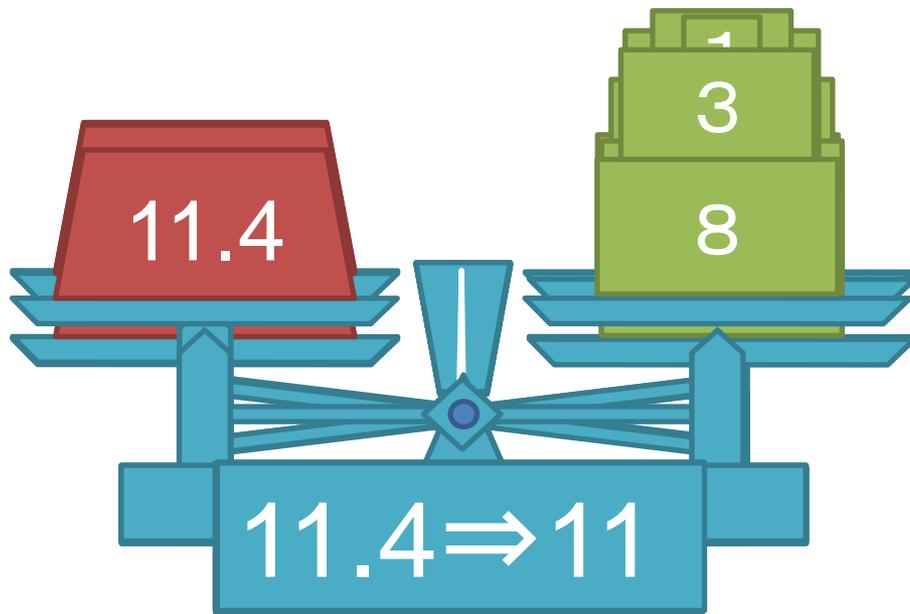
加算型補正

ECT-13-087

加算型

入力電圧11.4V参照電圧重み 8, 5, 3, 2, 1

値を加算するか加算しないかを定める方法



載せる (加算する) 1出力

8

3

載せない(加算しない) 0出力

5

2

1

デジタル出力

1 0 1 0 0



出力計算方法

$$8 + 0 + 3 + 0 + 0 = 11$$

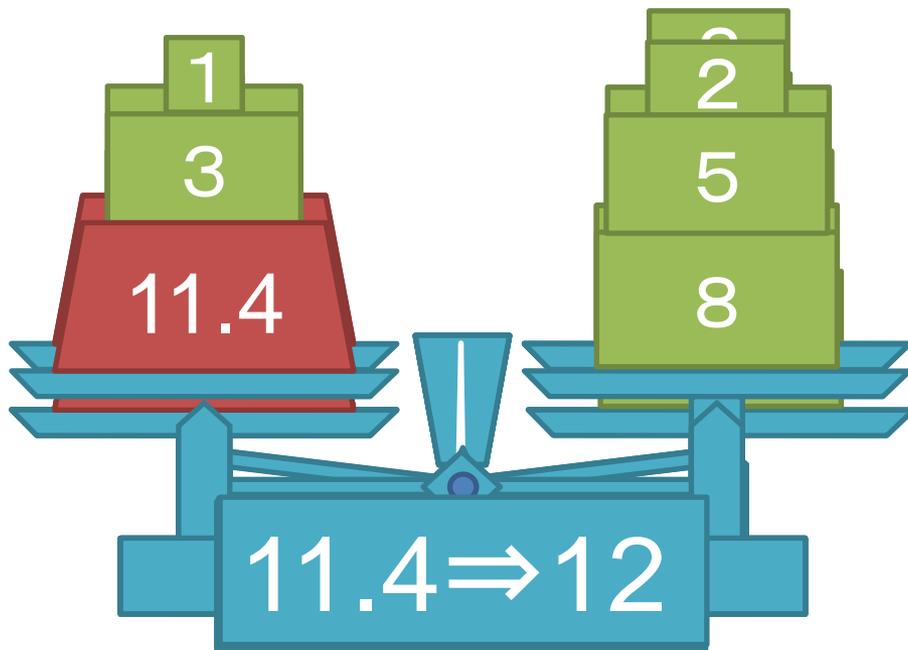
加減算型補正

ECT-13-087

加減算型

入力電圧11.4V参照電圧重み 8, 5, 3, 2, 1

値を加算するか減算するかを決める方法



右に載せる(加算する) 1 出力



左に載せる(減算する) 0 出力



デジタル出力

1 0 1 0 1

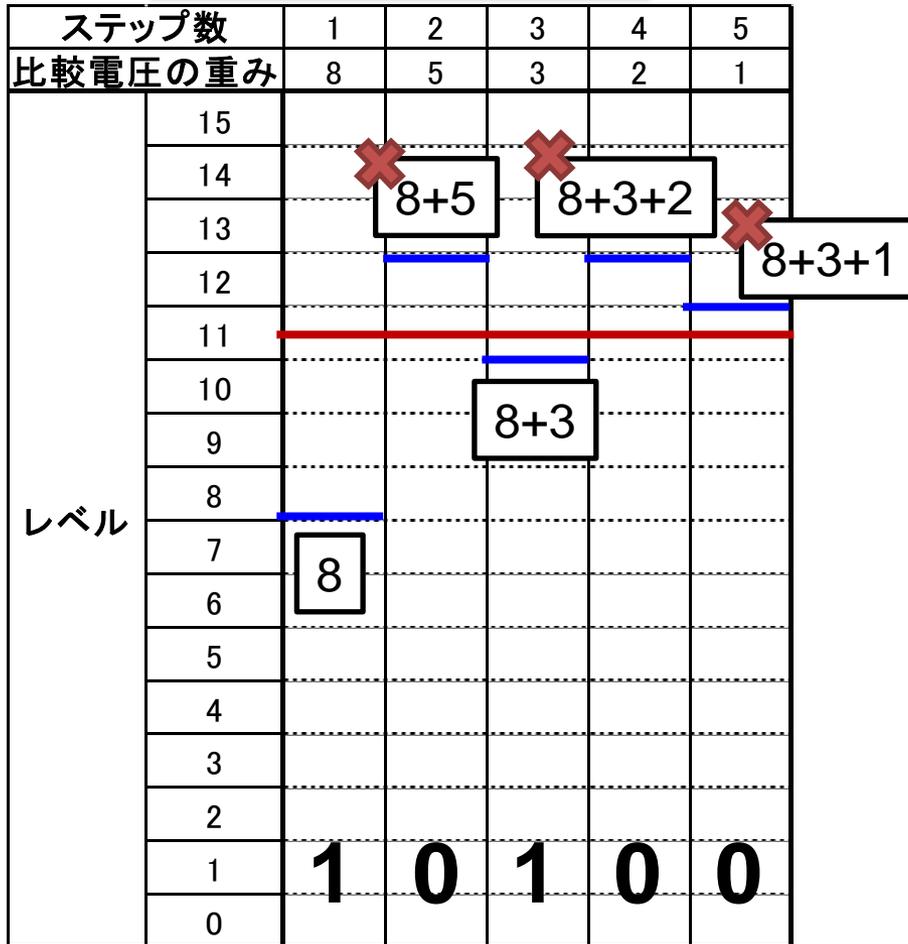
出力計算方法

$$8 + 5 - 3 + 2 - 1 + 0.5 - 0.5 = 11$$

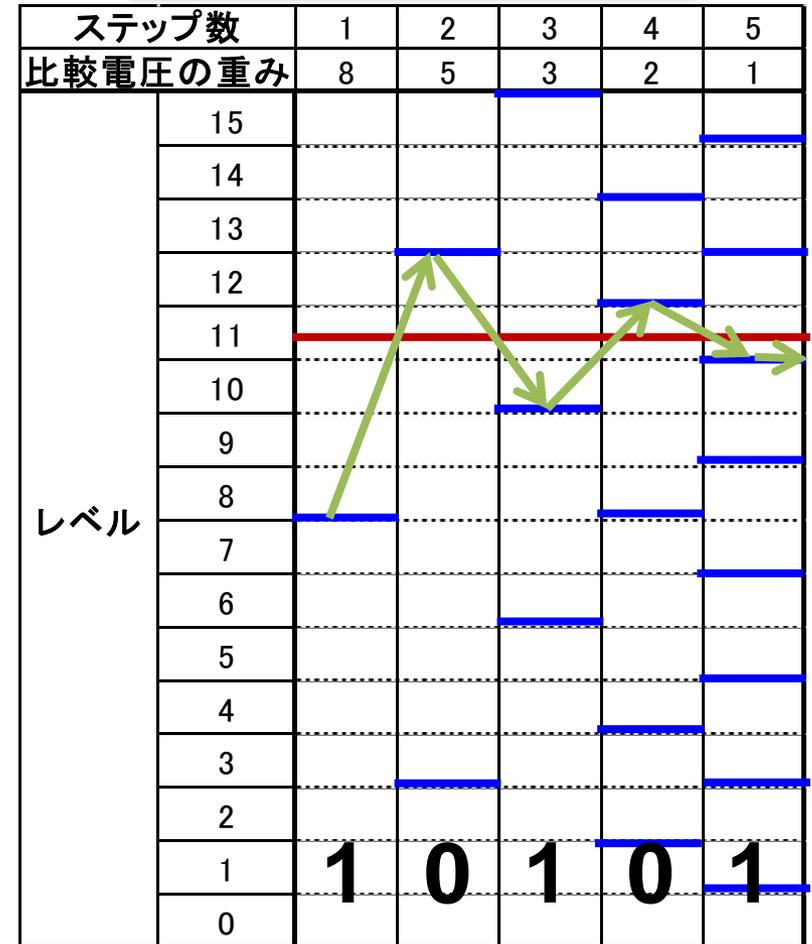
加算型と加減算型の比較

ECT-13-087

4bit5step加算型



4bit5step加減算型



これらの方法を利用するとき丁度良い重みはあるだろうか？₂₀

- 研究背景、目的
- 冗長性を持つSAR ADCについて
- **フィボナッチ数列を用いた冗長性設計**
- 理論シミュレーション結果
- 電流源DACへの応用
- まとめ、今後の課題

フィボナッチ数列とは

ECT-13-087

フィボナッチ数列は以下のように定義される数列のことである

$$F_0 = 0$$

$$F_1 = 1$$

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

方法③漸化式法

方法①フィボナッチ数法

初めの項を計算すると

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,
233, 377, 610, 987, 1596, 2583, 4180, 6764,
10945... ⇒ Fibonacci number と呼ばれる

また隣り合う項の比率は以下に収束する(約1.6進)

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

方法②黄金比法

収束比率 φ : 黄金比 (Golden ratio)

フィボナッチ数列をどのように使うか

フィボナッチ数列の性質
隣り合う項の比率は約1.6倍である



フィボナッチ数列は**整数**で**1.6進**を実現できる
※ADCの基準電圧は整数でなければいけない



基準電圧の重みに使うことで
2進に比べ**適度な冗長性設計**となる

冗長性を持つSAR ADC

参照電圧の大きさ⇒二進数より進みが遅い
すべて整数である

従来は参照電圧の大きさを開発者に委ねていた



通常、整数の小数倍は小数となってしまうが
フィボナッチ数列は**整数**で**1.6進**を実現できる



フィボナッチ数列を用いた参照電圧決定法を提案

デジタル表現値が複数パターン存在 ⇒ 誤差補正可能
先ほどの例 $(11)_{10} \Rightarrow 10100, 10011, 01111$

それぞれの値の表現パターン数の多さが
誤差補正力になっている



表現パターン数多い ⇨ 誤差補正力高い

重み付け理想

- 表現パターン数が多い
- 表現パターン数に偏在性がない
- ステップ数ができるべく少ない

フィボナッチ数列重み付け方法 ECT-13-087

重み(比較に使う基準電圧)の決定方法は以下の3つが考えられる

① フィボナッチ数法 Fibonacci number method

- フィボナッチ数をそのまま用いて1.6進を実現する方法

② 黄金比法 Golden ratio method

- 黄金比で割って1.6進を実現する方法

③ 漸化式法 Fibonacci recurrence relation method

- 漸化式の関係を利用して1.6進を実現する方法

※重みは整数のため小数は小数第一位で四捨五入する

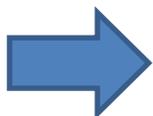
重み付け例①フィボナッチ数法 ECT-13-087

①フィボナッチ数法 Fibonacci number method

8bit分解能(256レベル)

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, ...

項合計376・12段階



8bit 12step

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

重み付け例②黄金比法

ECT-13-087

② 黄金比法

Golden ratio method

8bit分解能(256レベル)

黄金比 1.61803

$$256 \div 2 = 128 \Rightarrow 128$$

$$128 \div 1.61803 = 79.10 \Rightarrow 79$$

$$79.10 \div 1.61803 = 48.89 \Rightarrow 49$$

$$48.89 \div 1.61803 = 30.21 \Rightarrow 30$$

↓繰り返し

$$1.040 \div 1.61803 = 0.64 \Rightarrow 1$$

$$0.6432 \div 1.61803 = 0.39 \Rightarrow \times$$

12段階

8bit 12step

1, 1, 2, 3, 4, 7, 12, 19, 30, 49, 79, 128

重み付け例③漸化式法

ECT-13-087

③ 漸化式法

Fibonacci recurrence relation method

8bit分解能(256レベル)

$$\begin{array}{rclcl} 256 \div 2 & = & 128 & \Rightarrow & 128 \\ 128 \div 1.61803 & = & 80 & \Rightarrow & 80 \\ 128 - 80 & = & 48 & \Rightarrow & 48 \\ 80 - 48 & = & 32 & \Rightarrow & 32 \\ 48 - 32 & = & 16 & \Rightarrow & 16 \\ 32 - 16 & = & 16 & \Rightarrow & 16 \\ 16 - 16 & = & 0 & \Rightarrow & 0 \\ 16 - 0 & = & 16 & \Rightarrow & 16 \\ 0 - 16 & = & -16 & \Rightarrow & -16 \\ 16 - (-16) & = & 32 & \Rightarrow & 32 \end{array}$$

値が発散してしまう
表現できない
パターンがある

このままでは
使用できない

漸化式法の考察

ECT-13-087

項比率を調べてみると…

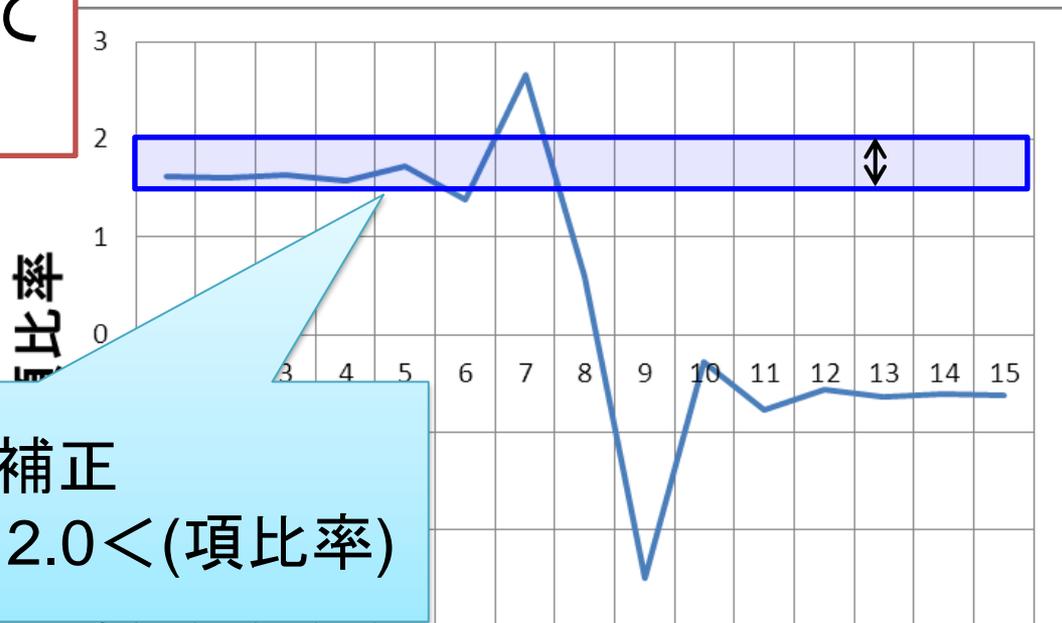
項の振動が起こる原因は

項比率が1.618よりもかなり小さくなる場所にある



項比率に条件を定めて
値を補正する

8bit項比率グラフ



項比率補正
(項比率) < 1.5 or 2.0 < (項比率)

重み付け例③漸化式法

ECT-13-087

8bit分解能(256レベル)

$$\begin{array}{llll} 256 \div 2 & = 128 & \Rightarrow 128 \\ 128 \div 1.61803 & = 80 & \Rightarrow 80 \\ 128 - 80 & = 48 & \Rightarrow 48 \\ 80 - 48 & = 32 & \Rightarrow 32 \\ 48 - 32 & = 16 & \Rightarrow 16 \\ 32 - 16 & = 16 & \Rightarrow 16 \\ \\ 16 \div 1.61803 & = 9.89 & \Rightarrow 10 \\ 16 - 10 & = 6 & \Rightarrow 6 \\ 10 - 6 & = 4 & \Rightarrow 4 \\ 6 - 4 & = 2 & \Rightarrow 2 \\ 4 - 2 & = 2 & \Rightarrow 2 \\ 2 \div 1.61803 & = 1.24 & \Rightarrow 1 \\ 2 - 1 & = 1 & \Rightarrow 1 \end{array}$$

項比率

1.60
1.67
1.50
2.00
1.00

補正対象
(項比率) < 1.5

補正

8bit 11step

1, 1, 2, 4, 6, 10,
16, 32, 48, 80, 128

8bitSAR ADC 重み付け結果まとめ

ECT-13-087

8bit分解能 (256レベル)において

①フィボナッチ数法

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 ⇒ 8bit12step 項合計376
冗長ステップ数4

②黄金比法

1, 1, 2, 3, 4, 7, 12, 19, 30, 49, 79, 128 ⇒ 8bit12step 項合計335
冗長ステップ数4

③漸化式法

1, 1, 2, 4, 6, 10, 16, 32, 48, 80, 128 ⇒ 8bit11step 項合計328
冗長ステップ数3

これらの重みを使って各レベルの表現パターン数を求める

3つの方法を比較する基準を考える

平均表現パターン

表現パターンが多いほど
誤判定を補正できる力が強いと言えるため
重要な評価項目である（ステップ数に依存する）

冗長使用率

以下の定義式で無駄の少なさを評価する
ステップ数が1増えるごとに理論上の
表現方法は2倍になるのでこれを利用して評価を行う

$$\text{冗長使用率(\%)} = \frac{\text{平均表現パターン数}}{2^N}$$

N : 冗長ステップ数

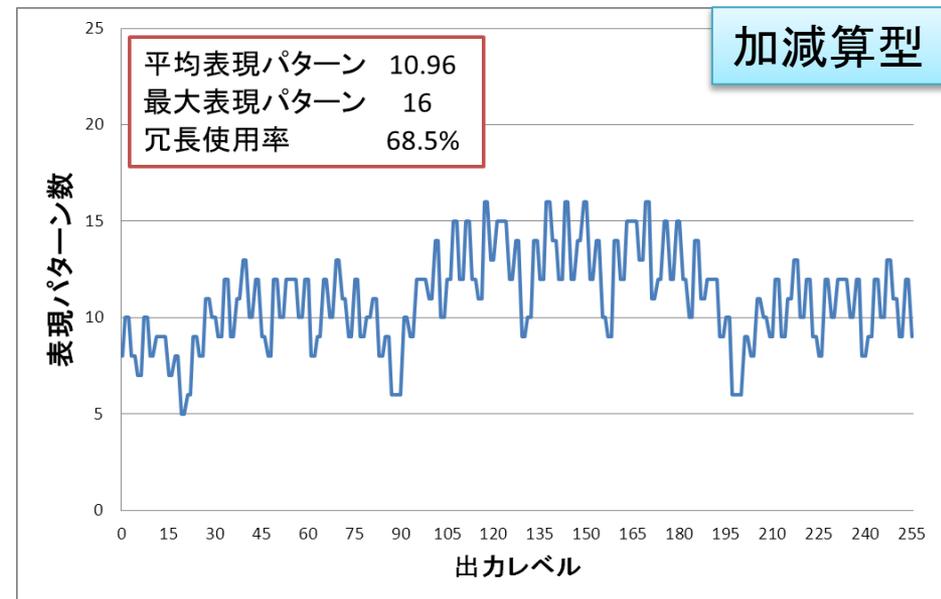
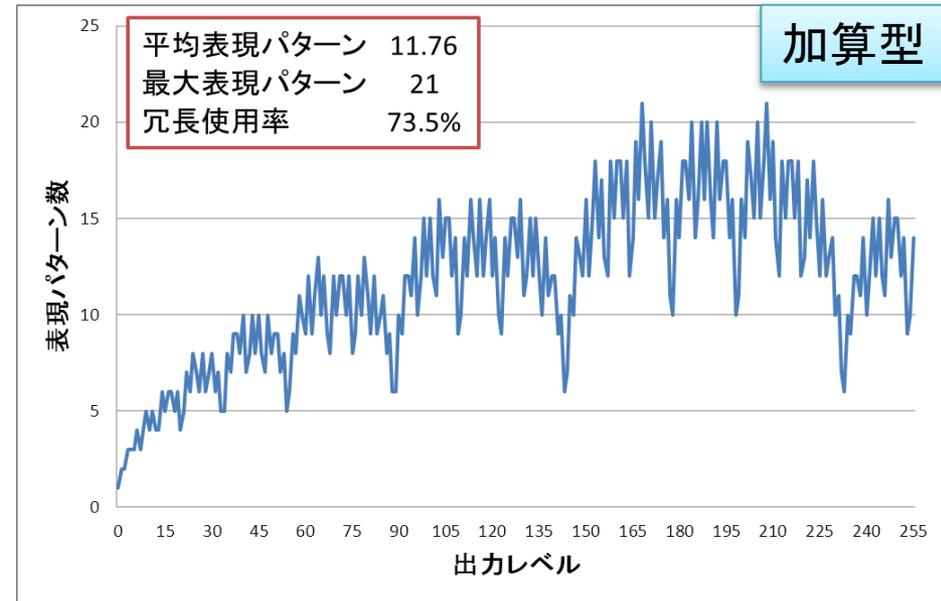
表現パターン数①方式

ECT-13-087

①フィボナッチ数法

8bit12step 項合計376

- 導出が単純である
- フィボナッチ数そのままであるので最も忠実に1.6進を実現する



表現パターン数②方式

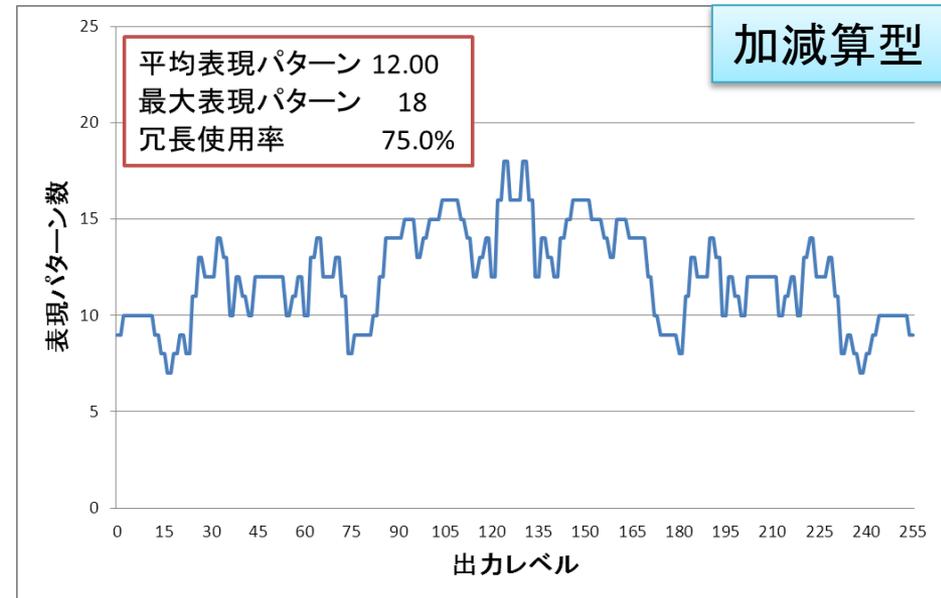
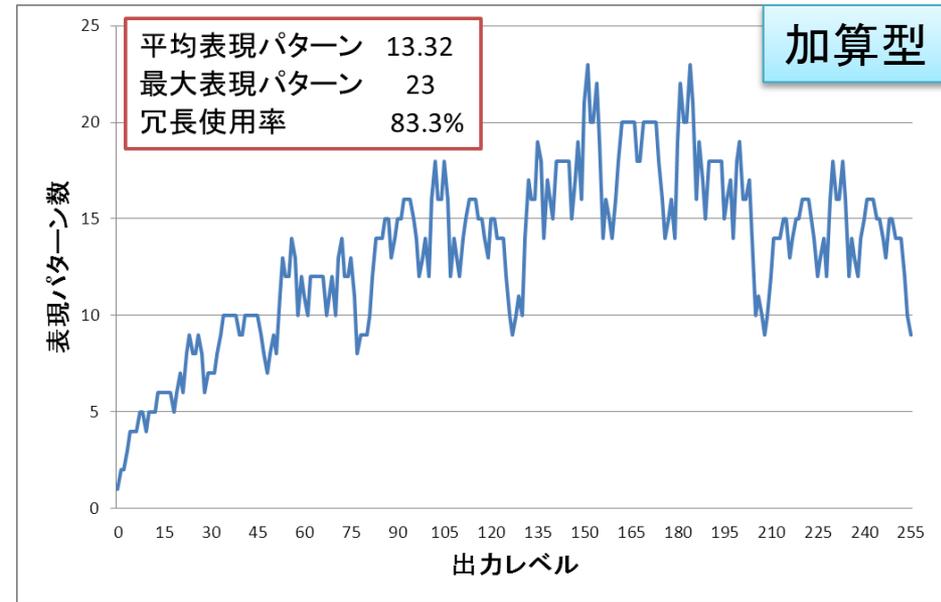
ECT-13-087

②黄金比法

8bit12step 項合計335

- 冗長ステップ数が大きくなる傾向
- 導出が単純である

※bit数によって1の項が3項出るが
2項に変更する



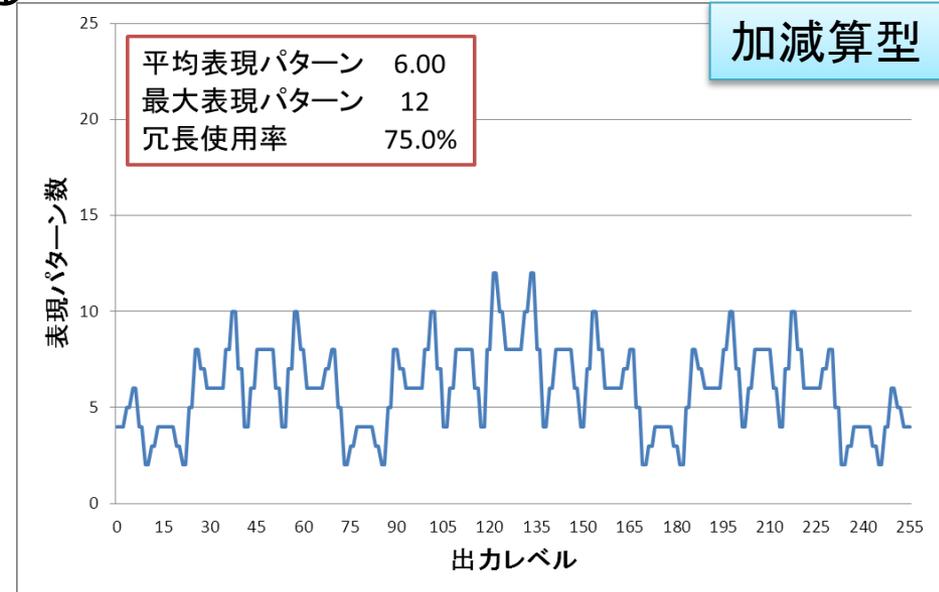
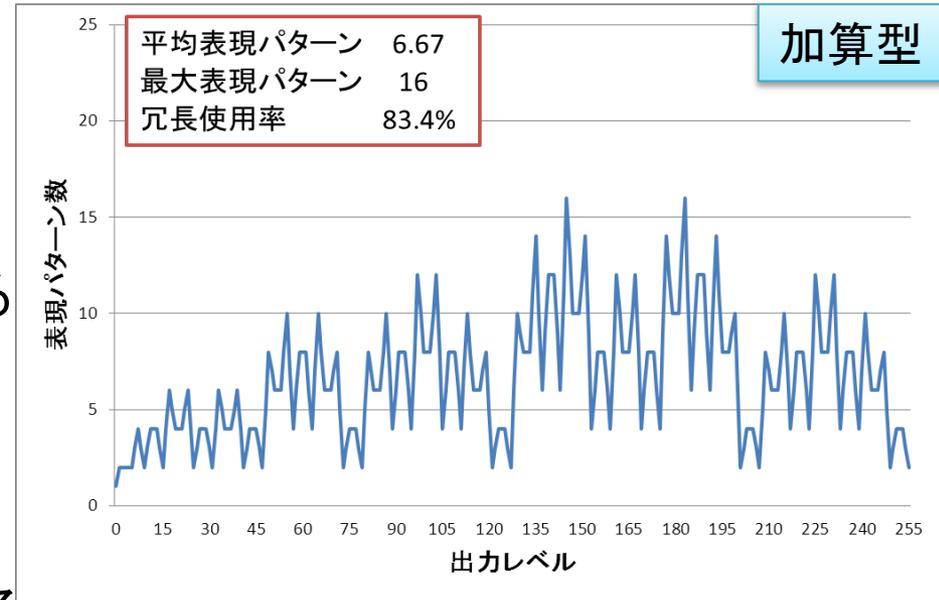
表現パターン数③方式

ECT-13-087

③漸化式法

8bit11step 項合計328

- 冗長ステップ数が小さくなる傾向がある
- 導出が複雑である
- 四捨五入はほとんど使わないために
フィボナッチ数列に近い項を出力できる
- 表現パターンに偏在性がみられる

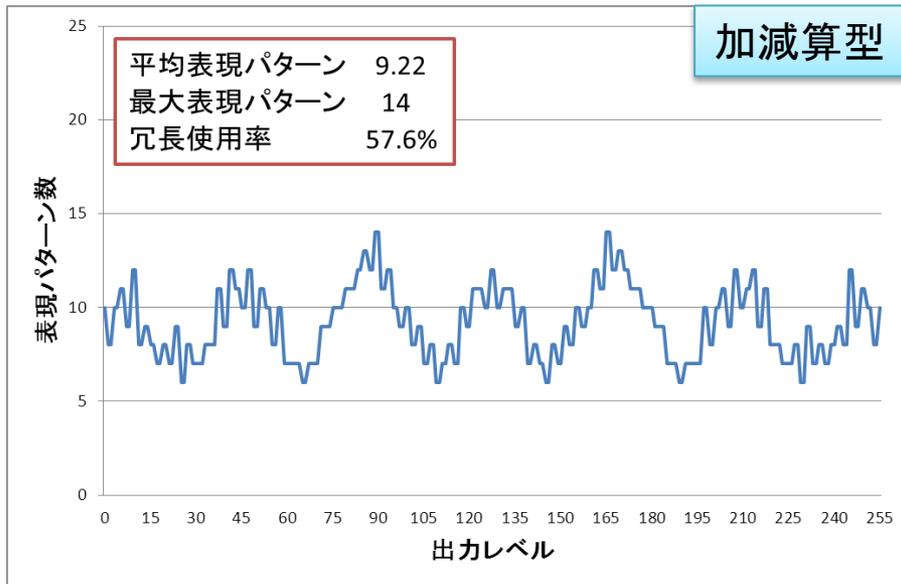


各方法の比較

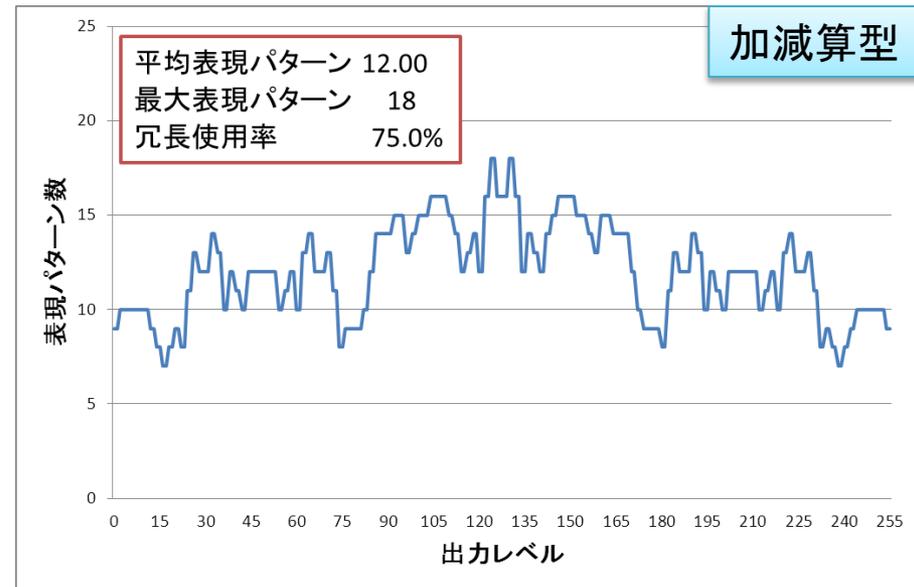
ECT-13-087

従来法(自由に定めた値を使う場合)と比較する

④従来法(12step)⇒1,2,3,5,7,14,24,42,62,80,102,128



従来法



黄金比法

◆ 加算型と加減算型の比較

- ・最大表現パターンや冗長使用率は
加算型の方が多くなる

◆ bit数増大に対する考察

- ・bit数の増加でbit数と冗長ステップ数の比率が減少

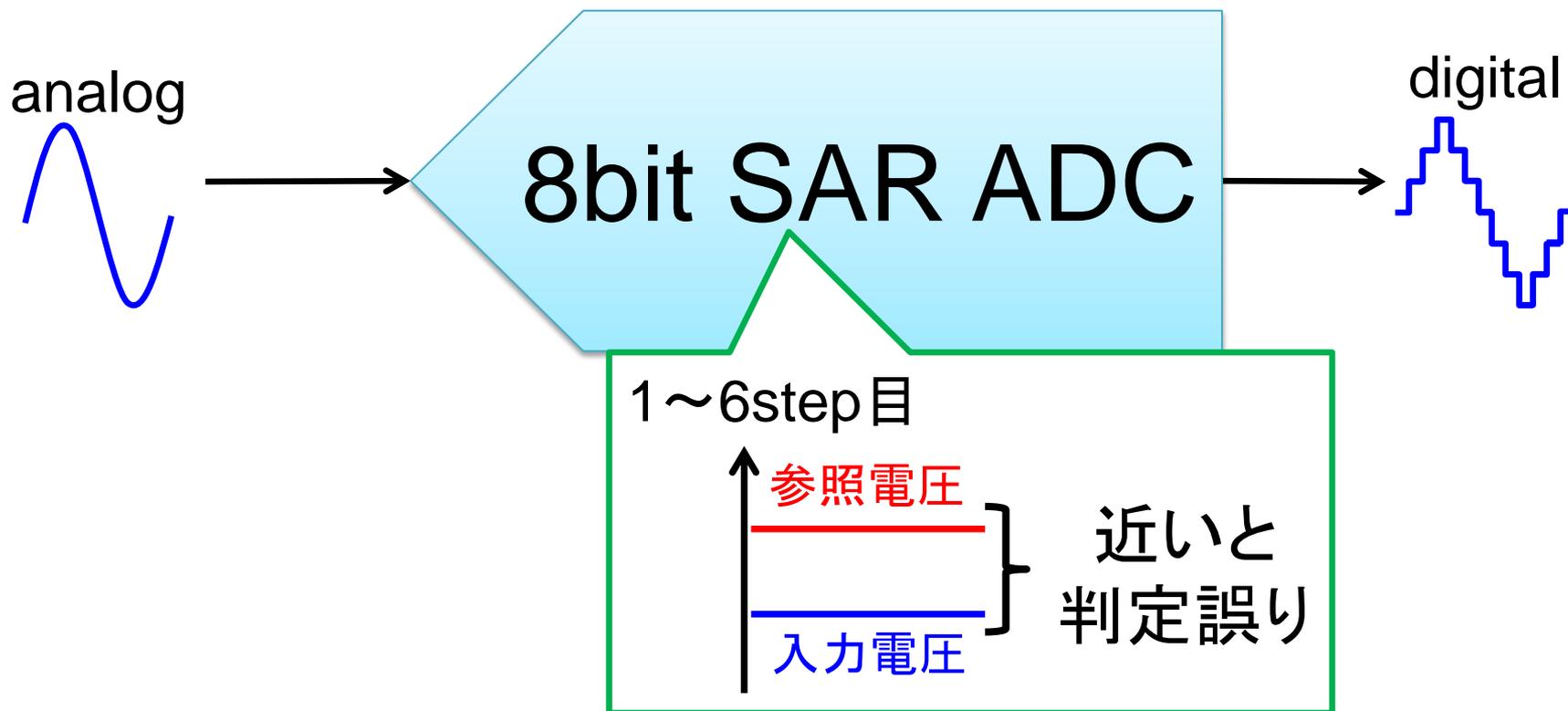
◆ 3方式比較

- ・どの方法も1. 6進を実現することができる
- ・ここでの考察だけではどの方法が適するかは
一概に言うことはできない

- 研究背景、目的
- 冗長性を持つSAR ADCについて
- フィボナッチ数列を用いた冗長性設計
- **理論シミュレーション結果**
- 電流源DACへの応用
- まとめ、今後の課題

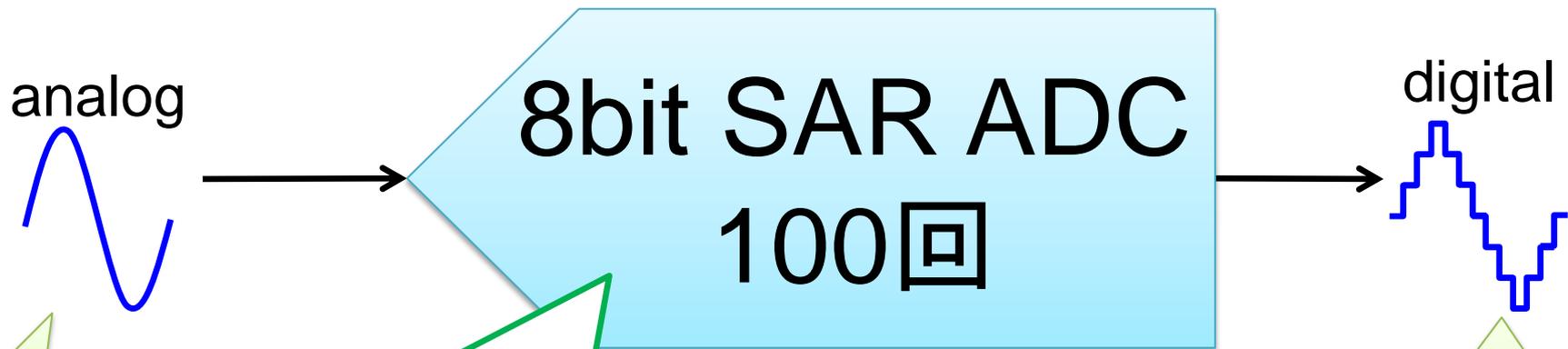
シミュレーションの想定

- ・ノイズのある環境で正弦波の8bitAD変換を行う
- ・参照電圧と入力電圧が近い値のときのみ
コンパレータが判定を誤る可能性がある
- ・後段では時間をかけて確実な判定をするものとする



シミュレーション条件

- ・C言語による8bit SAR ADC理論シミュレーション
- ・フルスケール正弦波入力、変換回数100回
- ・6step目まで判定を誤る可能性を持つ (rand関数使用)
⇒入力との差10以内で40%の誤り確率



0~255

1~6step目

↑ 参照電圧

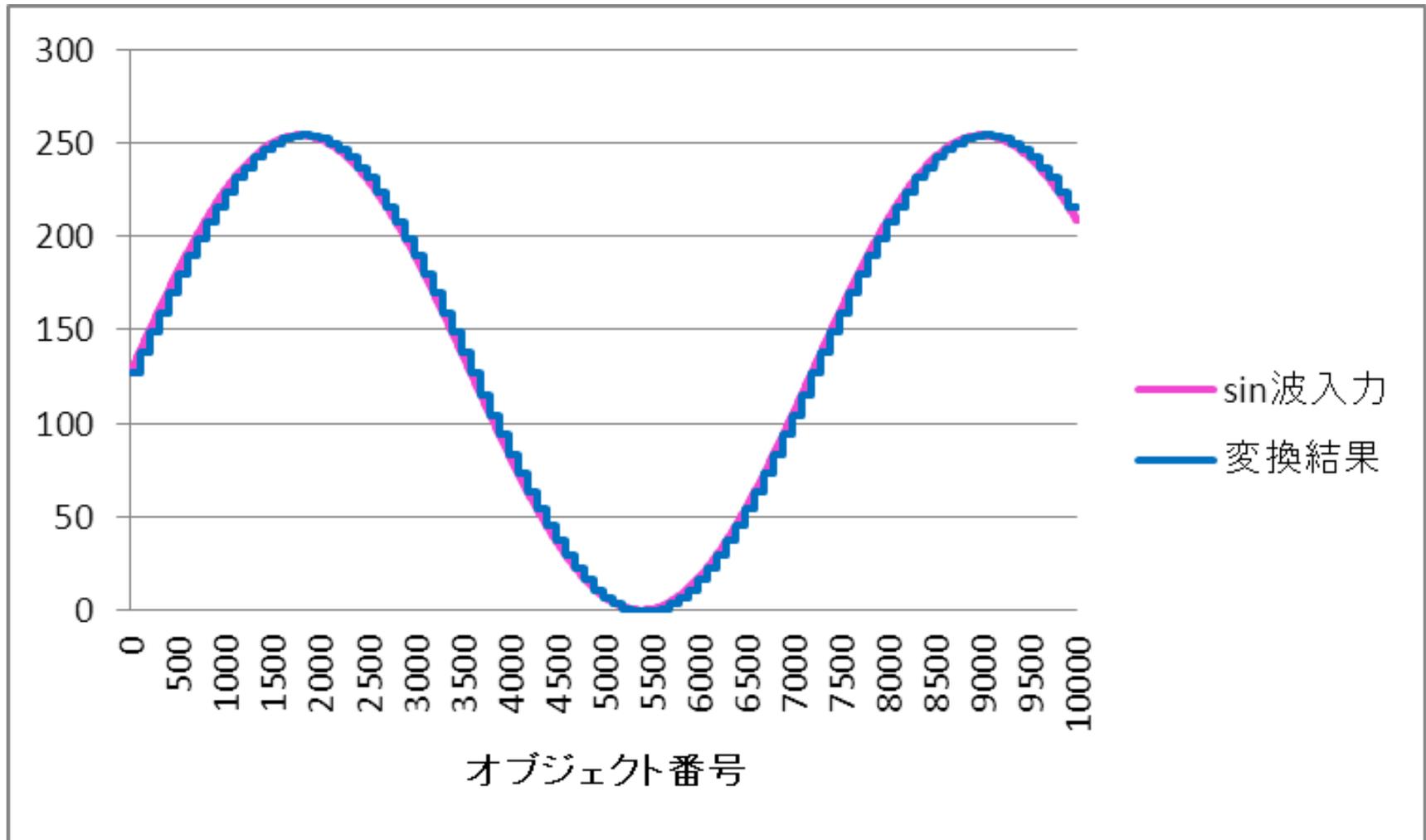
↑ 入力電圧

差が10以内
40%判定誤り

どれだけ誤差補正
できているか

誤判定なしAD変換

ECT-13-087



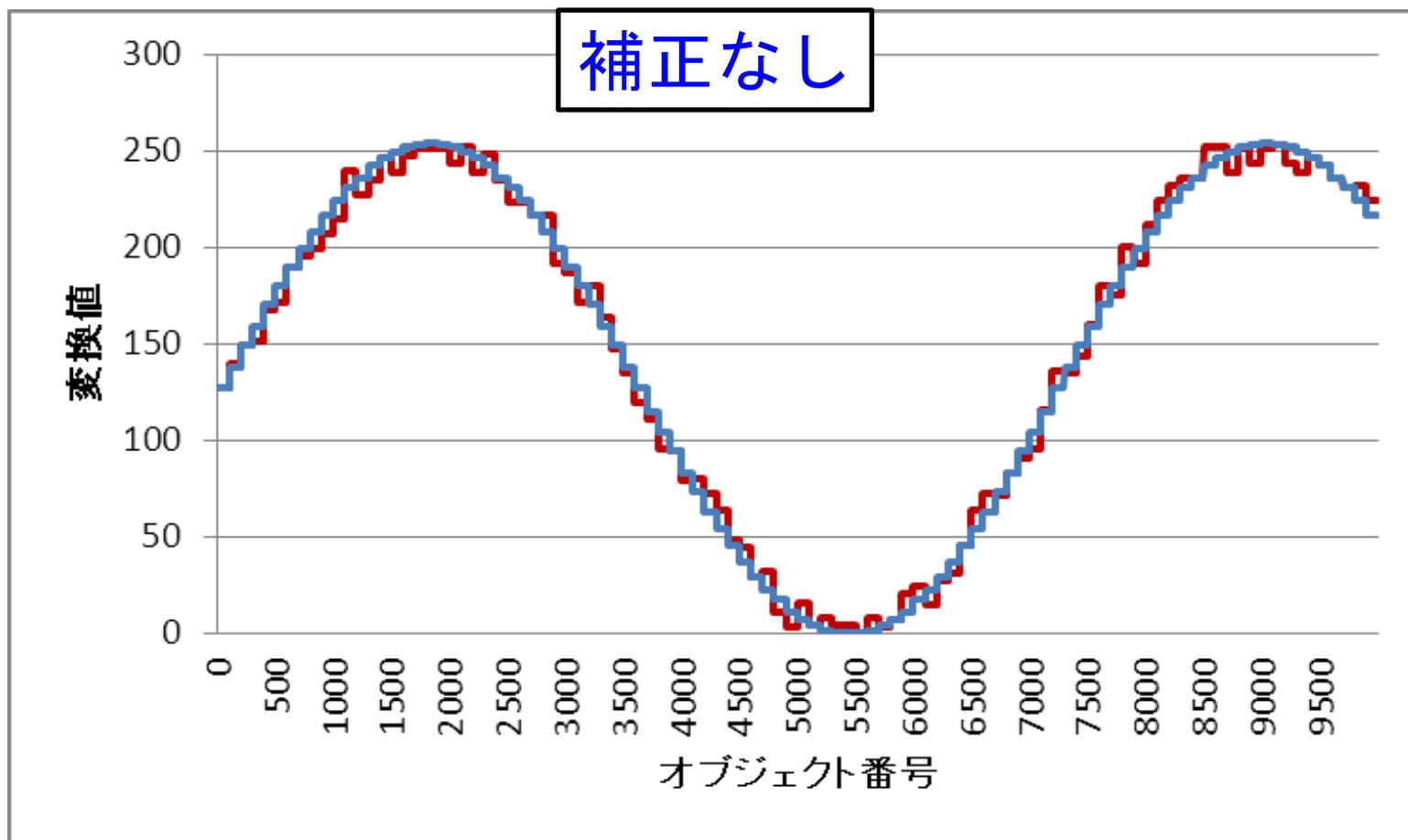
誤判定なしで正しく変換できている⇒比較対象に使う

判定誤りシミュレーション

ECT-13-087

判定1～6ステップ目の間

入力電圧と参照電圧の差10V以内で40%判定誤り



加算型デジタル誤差補正

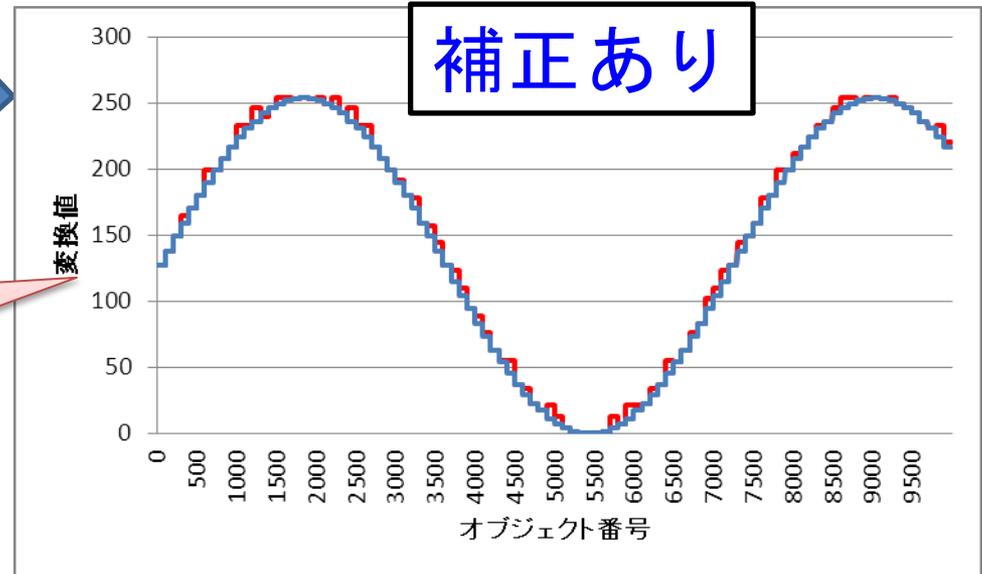
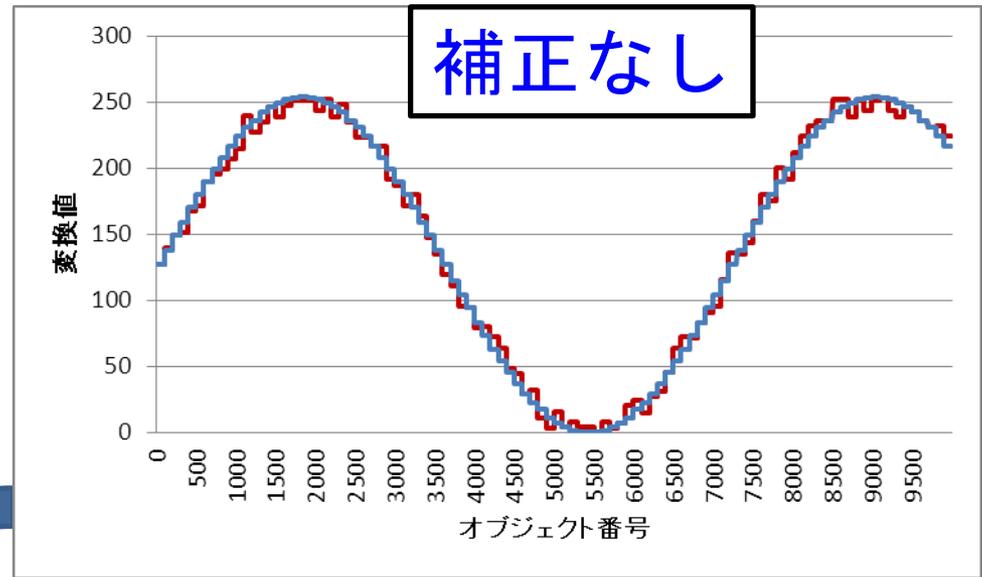
ECT-13-087

step数を増加
デジタル誤差補正を行う

①フィボナッチ数法
8bit12step
加算方式

加算方式
デジタル誤差補正

全体で46.0%
誤差を減少



加減算型デジタル誤差補正

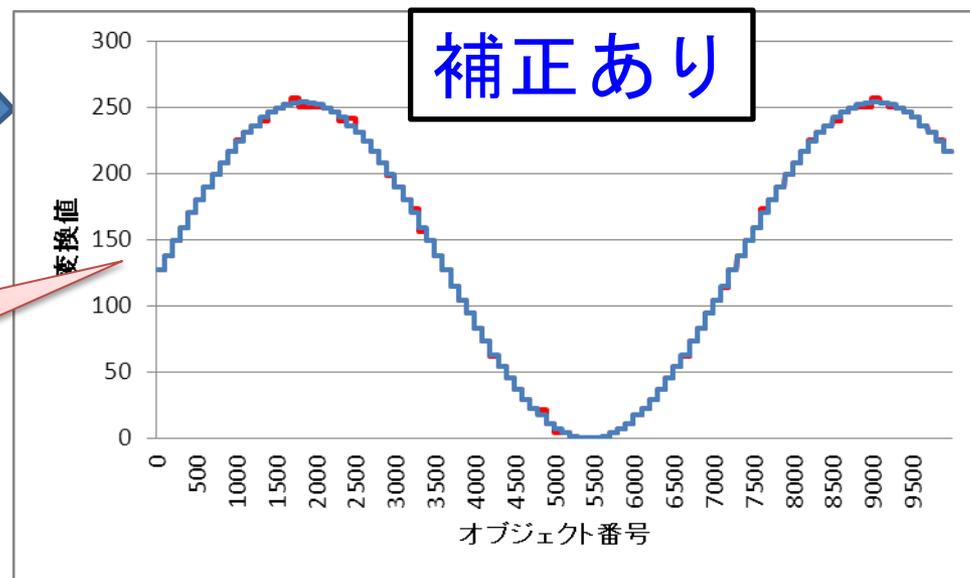
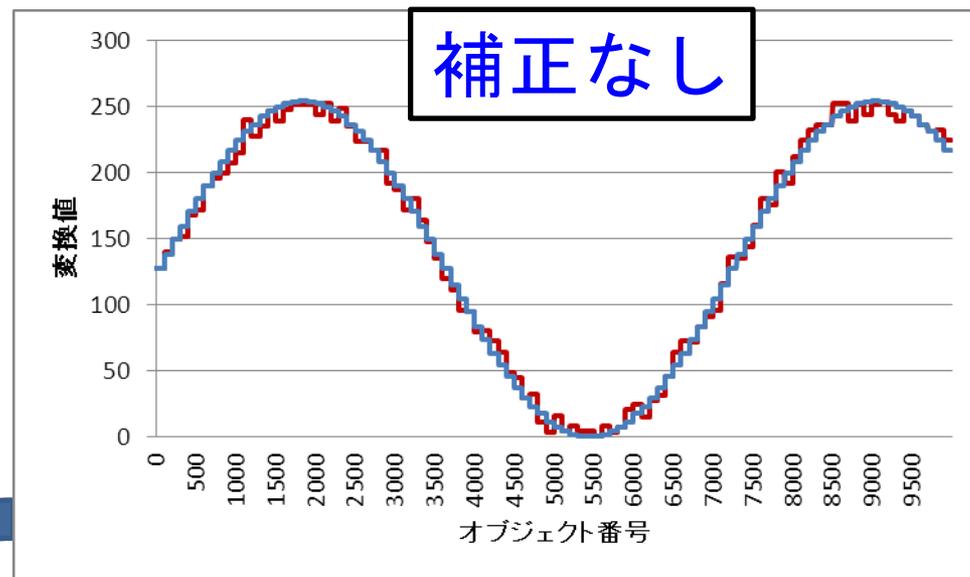
ECT-13-087

step数を増加
デジタル誤差補正を行う

①フィボナッチ数法
8bit12step
加減算方式

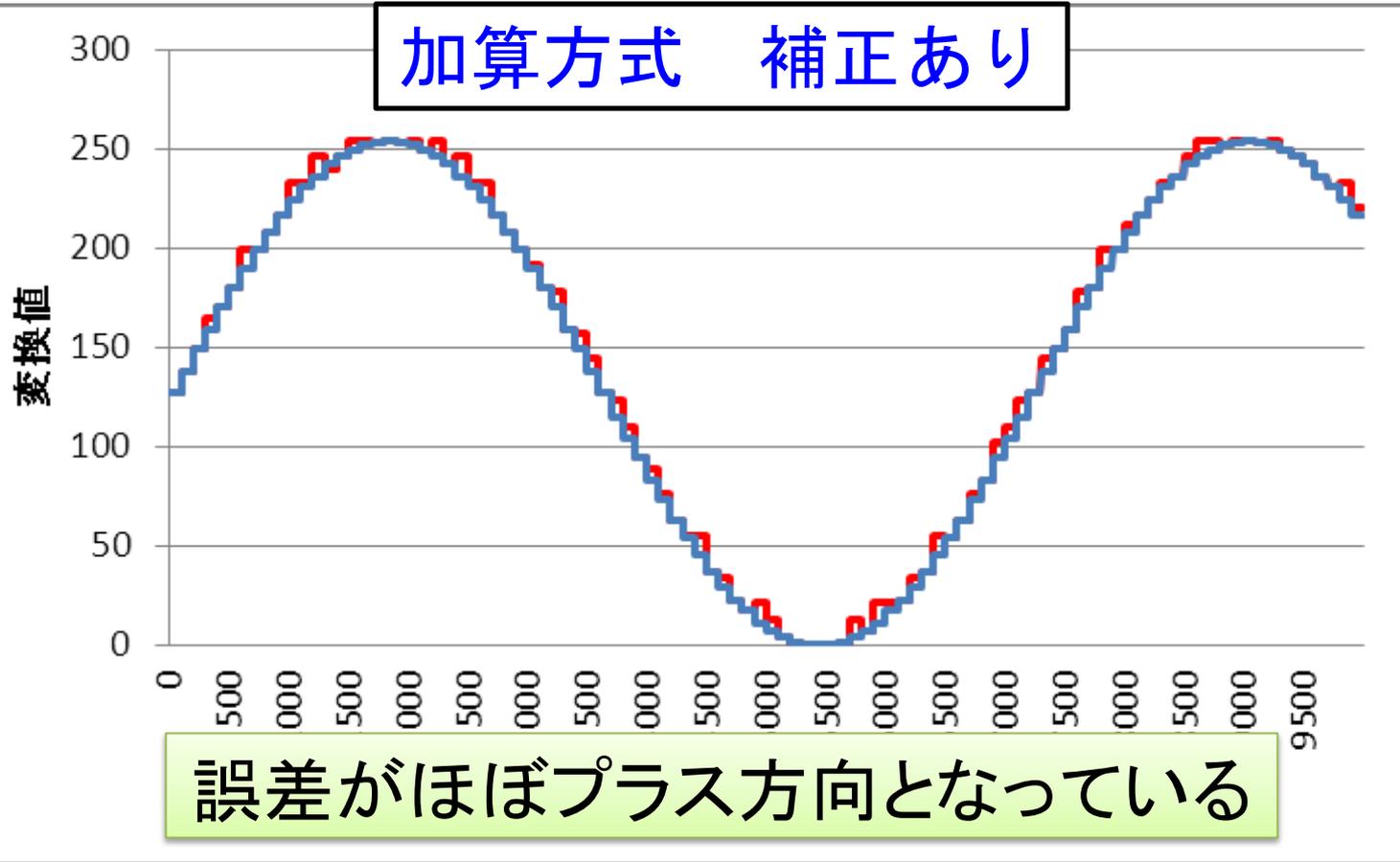
加減算方式
デジタル誤差補正

全体で89.0%
誤差を減少



各方式補正結果

ECT-13-087



どちら

50%程度までしか補正ができない

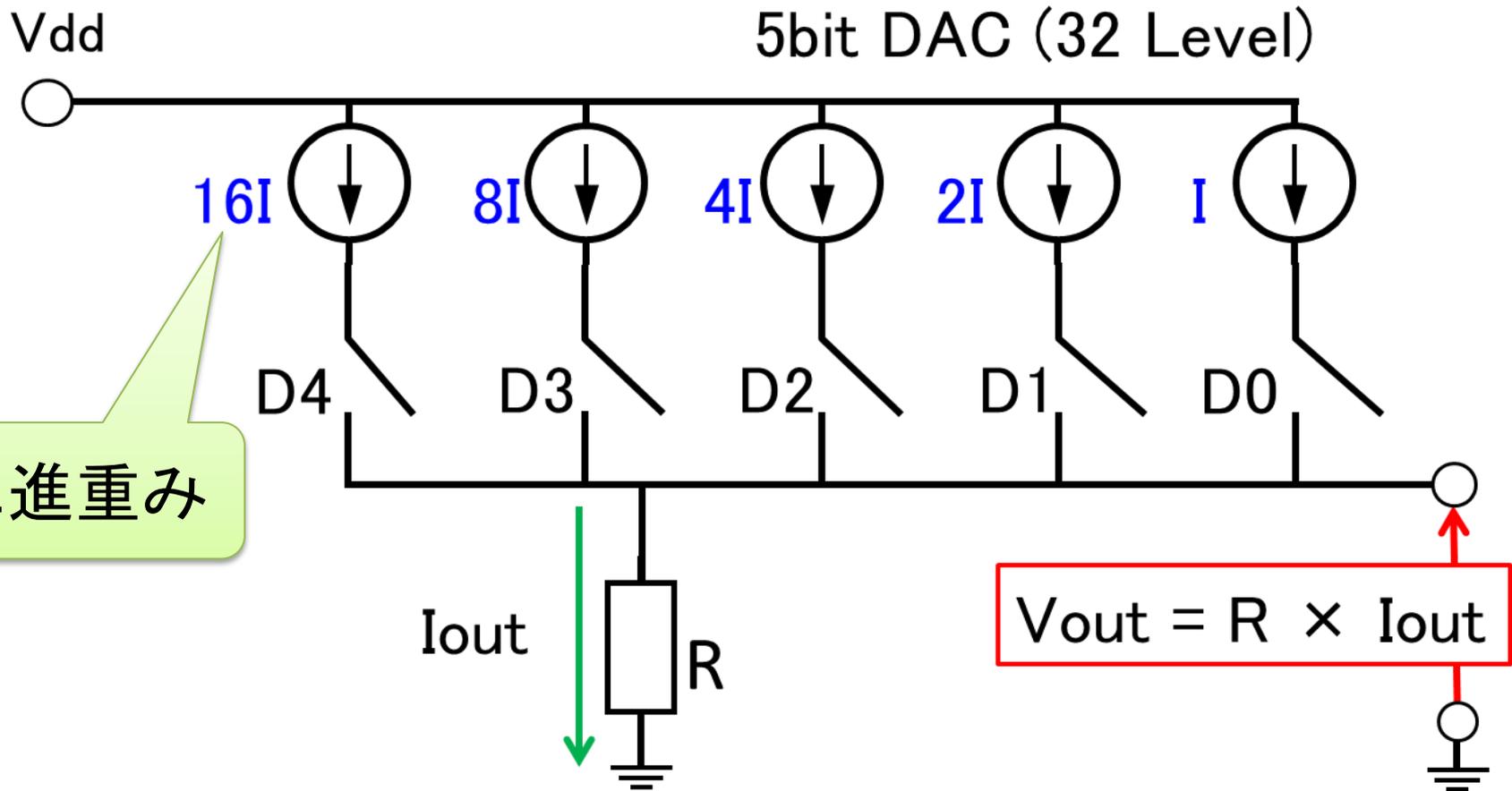
- 加減算方式がデジタル誤差補正に向けた方法である

- 研究背景、目的
- 冗長性を持つSAR ADCについて
- フィボナッチ数列を用いた冗長性設計
- 理論シミュレーション結果
- **電流源DACへの応用**
- まとめ、今後の課題

電流源DACとは

ECT-13-087

スイッチによって電流量を定め、抵抗を使い電圧を取り出す



二進重み

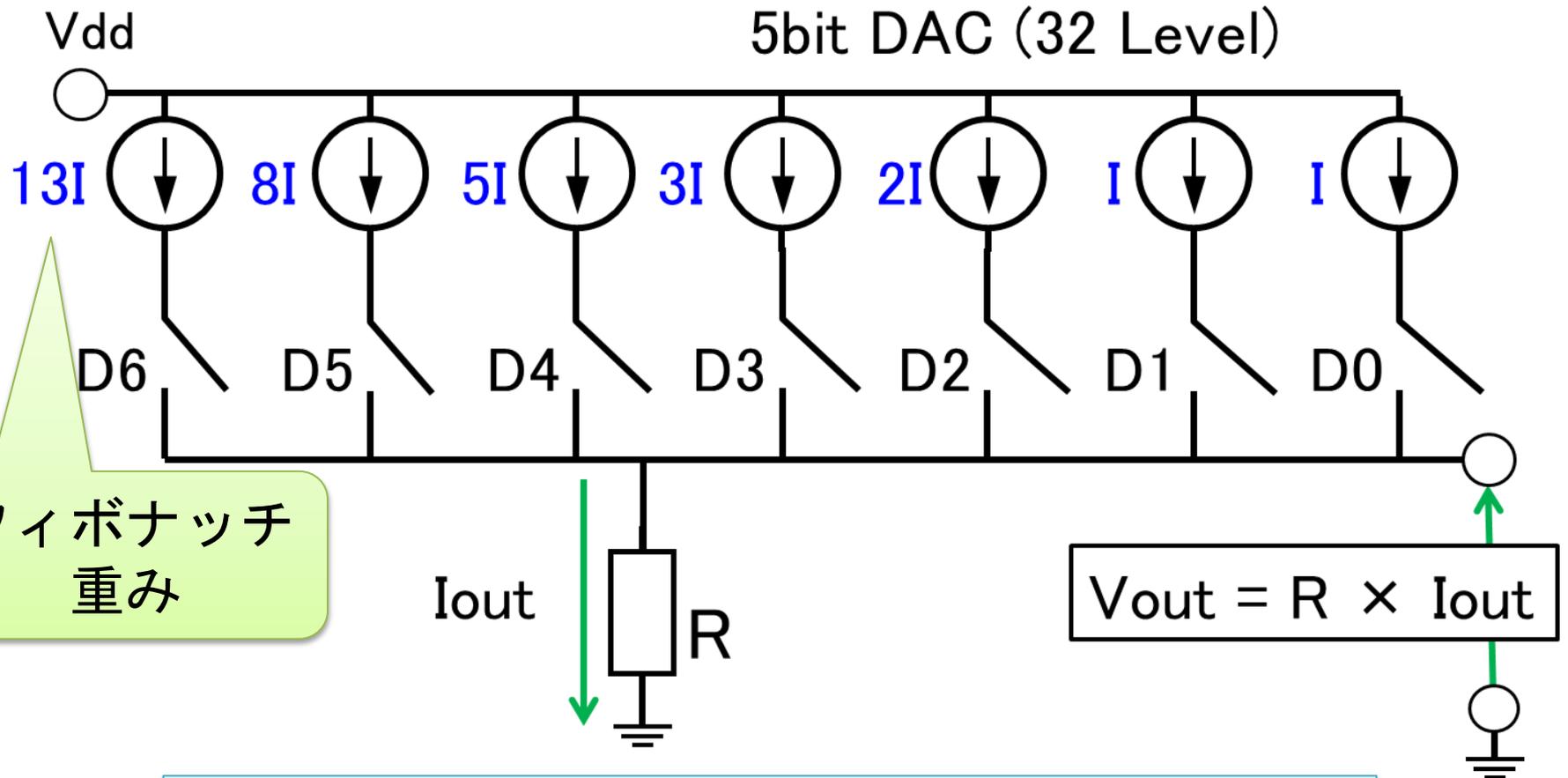
フィボナッチ数列を適用する

フィボナッチ数列を用いたDAC

ECT-13-087

フィボナッチ数列を電流源の重み付けに利用する

DACでは空間の冗長性(回路規模)を利用



DA変換器の高精度化理論を3種類検討!

フィボナッチ数列を用いたDA変換器の高精度化アーキテクチャ

① 冗長性の利用

一つのデジタル入力に対して、
複数の「オンになる電流源の組」が存在

A. 最も線形になる「オンになる電流源の組」を選択
⇒ 静的線形性の向上

B. ランダムに「オンになる電流源の組」を選択
⇒ SFDRの向上

② $F_{n+2} = F_n + F_{n+1}$ の関係式利用

下位2つの電流源の和が次の電流源になることを利用
⇒ 電流源を比較・校正

フィボナッチ数列を用いたDA変換器の高精度化アーキテクチャ

① 冗長性の利用

一つのデジタル入力に対して、
複数の「オンになる電流源の組」が存在

A. 最も線形になる「オンになる電流源の組」を選択
⇒ 静的線形性の向上

B. ランダムに「オンになる電流源の組」を選択
⇒ SFDRの向上

② $F_{n+2} = F_n + F_{n+1}$ の関係式利用

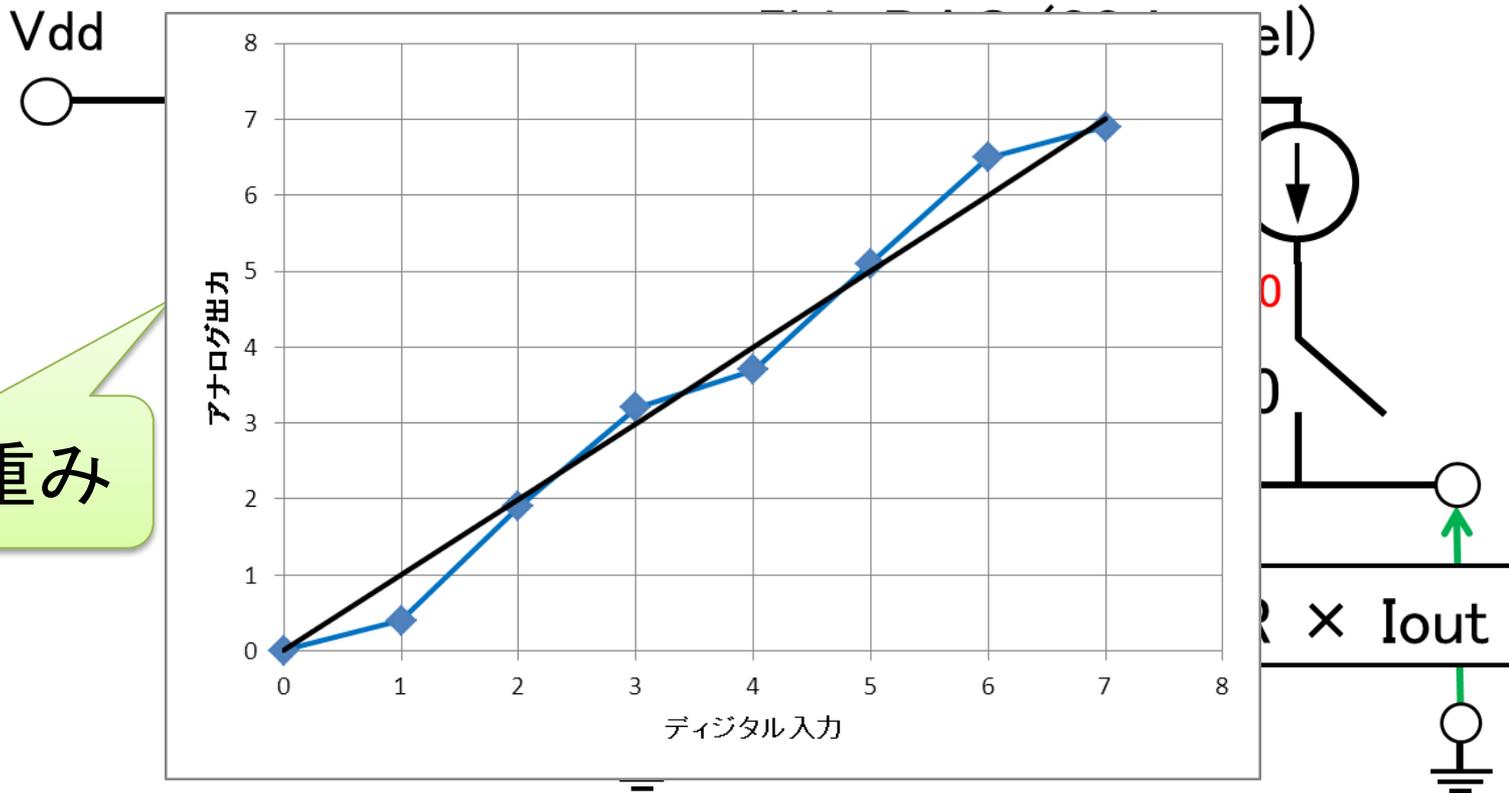
下位2つの電流源の和が次の電流源になることを利用
⇒ 電流源を比較・校正

電流源DACのミスマッチの影響

ECT-13-087

電流源にはそれぞれにばらつきが存在する

二進重み



アナログ値表現が一通りのみ
ミスマッチの影響をそのまま出力



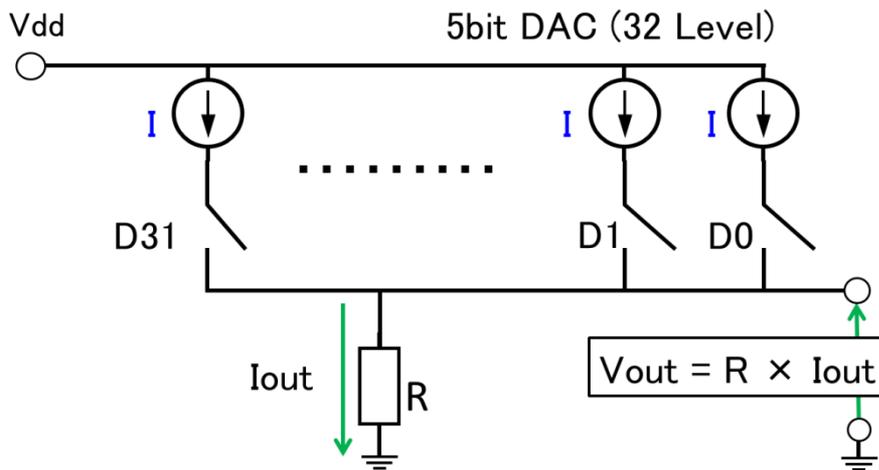
線形性の劣化

線形性向上の従来法

ECT-13-087

二進重み付けDAC(冗長性なし・回路規模小)

線形性向上 → セグメント型DAC(冗長性大・回路規模大)



電流源大きさ
すべて1I

→ 上位セグメント型+下位二進(冗長性中・回路規模中)

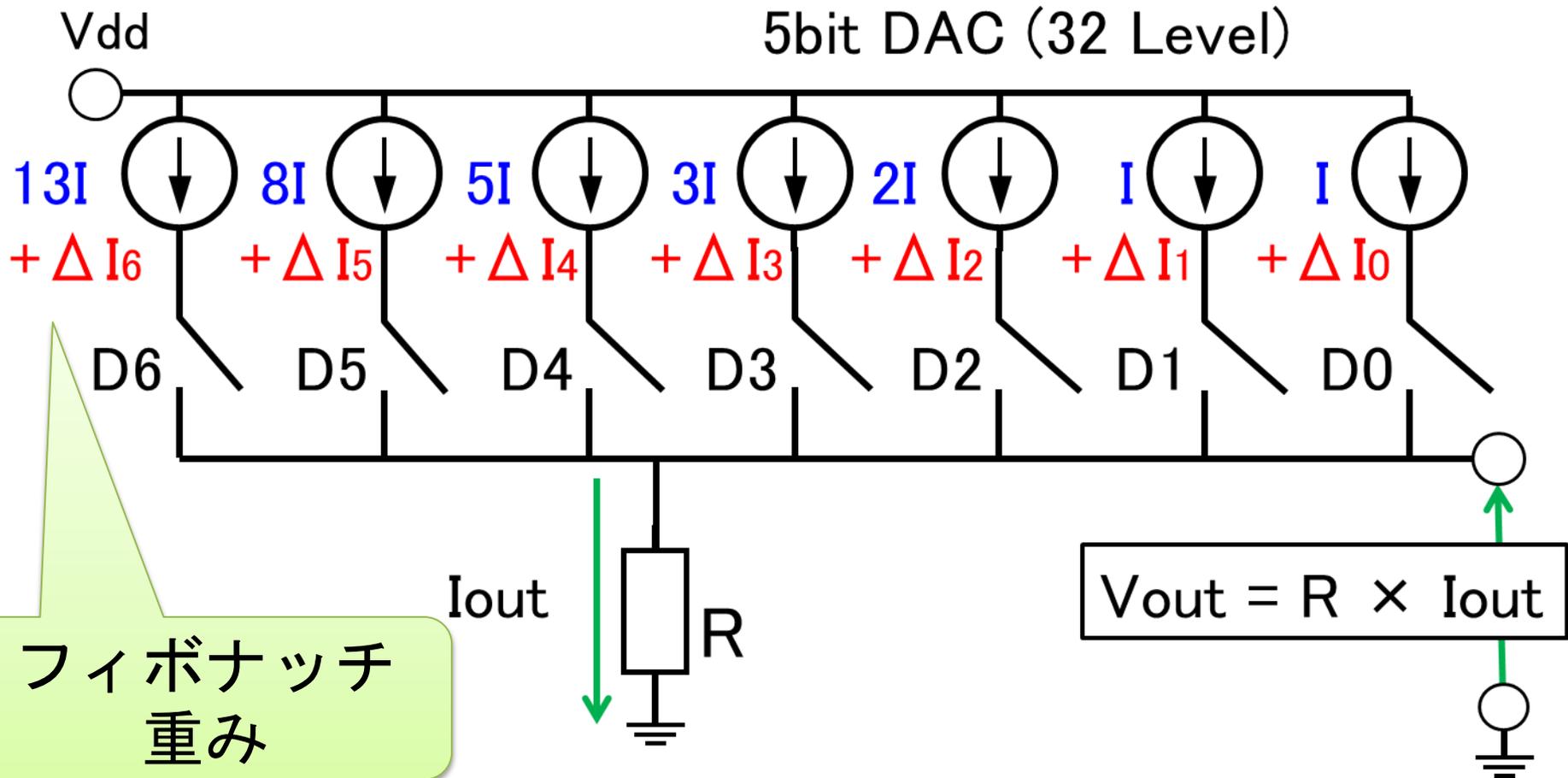


適度な冗長性のフィボナッチDACアーキテクチャ

電流源ミスマッチの考慮

ECT-13-087

電流源にはそれぞれにばらつきが存在する



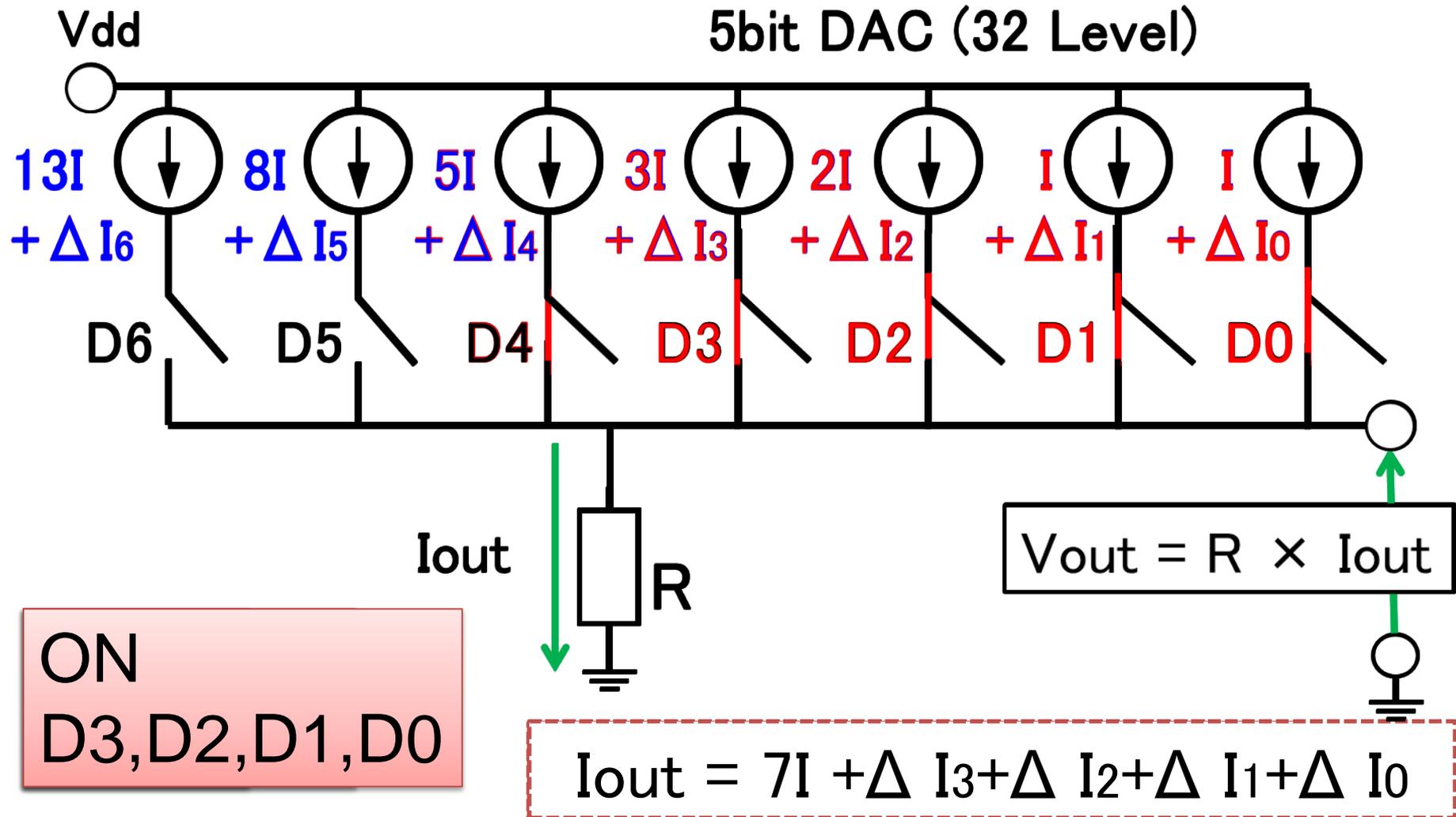
フィボナッチ
重み

冗長性によって改善

例) 冗長DAC変換

ECT-13-087

7Iを出力する場合



7Iを出力する場合

3通りの選択肢

$$\textcircled{1} I_{out} = 7I + \Delta I_4 + \Delta I_2$$

$$\textcircled{2} I_{out} = 7I + \Delta I_4 + \Delta I_1 + \Delta I_0$$

$$\textcircled{3} I_{out} = 7I + \Delta I_3 + \Delta I_2 + \Delta I_1 + \Delta I_0$$

3通りの I_{out} の中から
最も7Iに近い値を採用することで
電流のばらつきを低減可能！

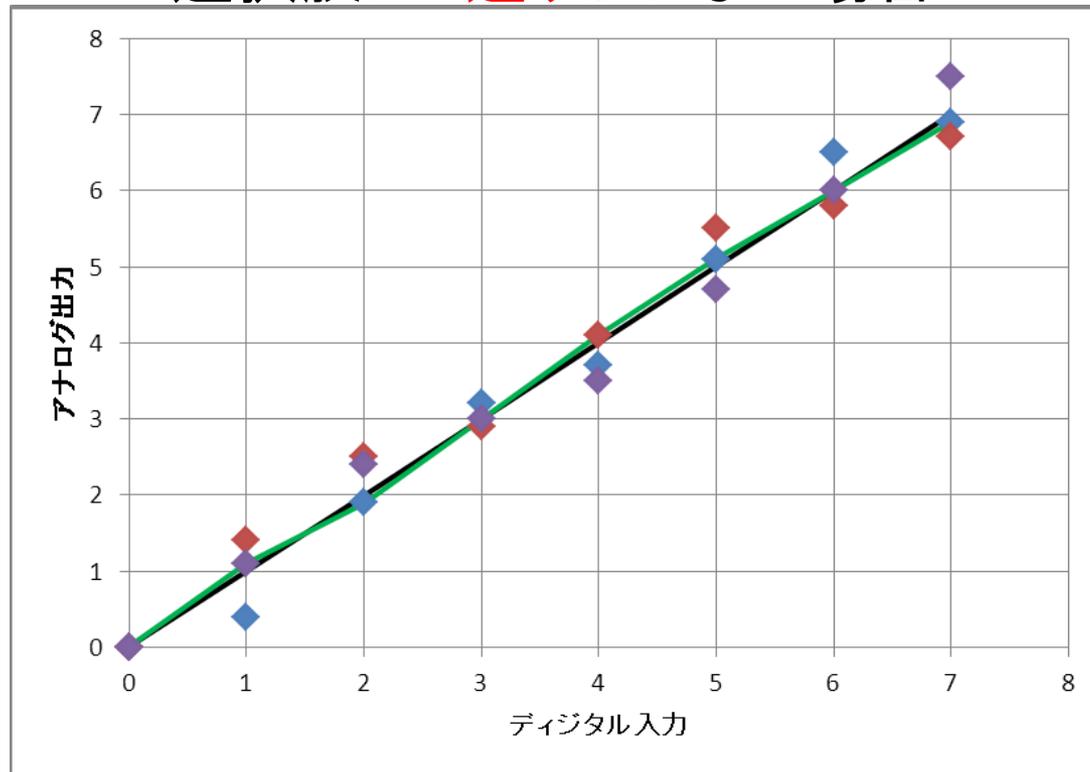


これをすべての値で行う

線形性向上の考え方

ECT-13-087

選択回路が**通通り**ある場合



冗長性を利用して複数のパターンから最適値を選択



電流ばらつきを抑え、**線形性を向上**できる！

フィボナッチ数列を用いたDA変換器の高精度化アーキテクチャ

① 冗長性の利用

一つのデジタル入力に対して、
複数の「オンになる電流源の組」が存在

A. 最も線形になる「オンになる電流源の組」を選択
⇒静的線形性の向上

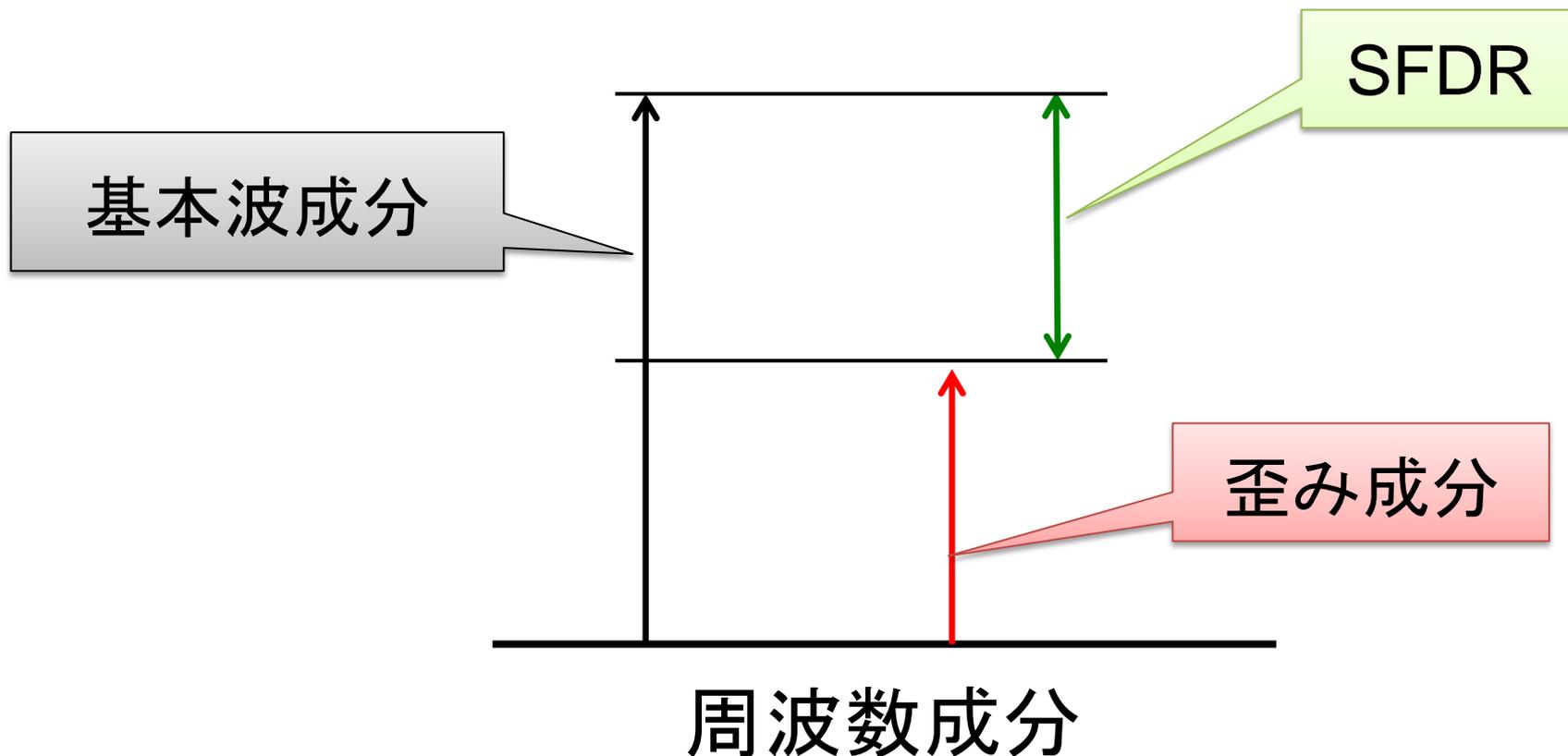
B. ランダムに「オンになる電流源の組」を選択
⇒SFDRの向上

② $F_{n+2} = F_n + F_{n+1}$ の関係式利用

下位2つの電流源の和が次の電流源になることを利用
⇒電流源を比較・校正

DA変換器の性能評価指標

SFDR: sin波をAD変換しDA変換すると生じる
周波数スペクトルと基本波信号の比



SFDRの向上

ECT-13-087

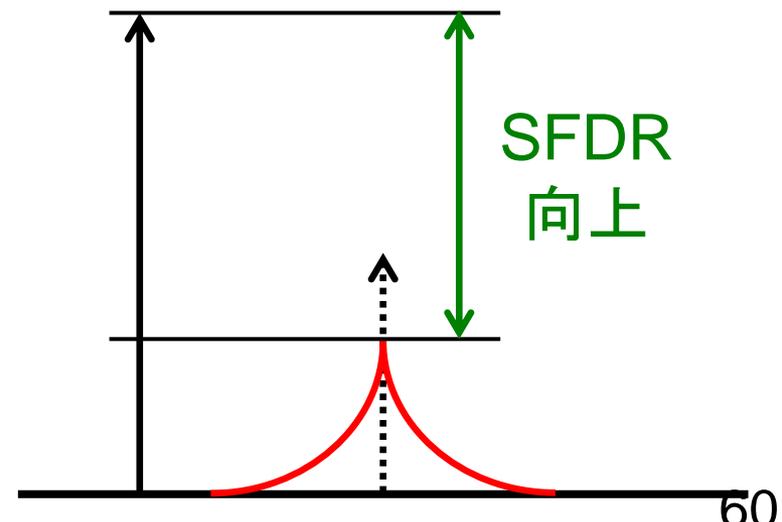
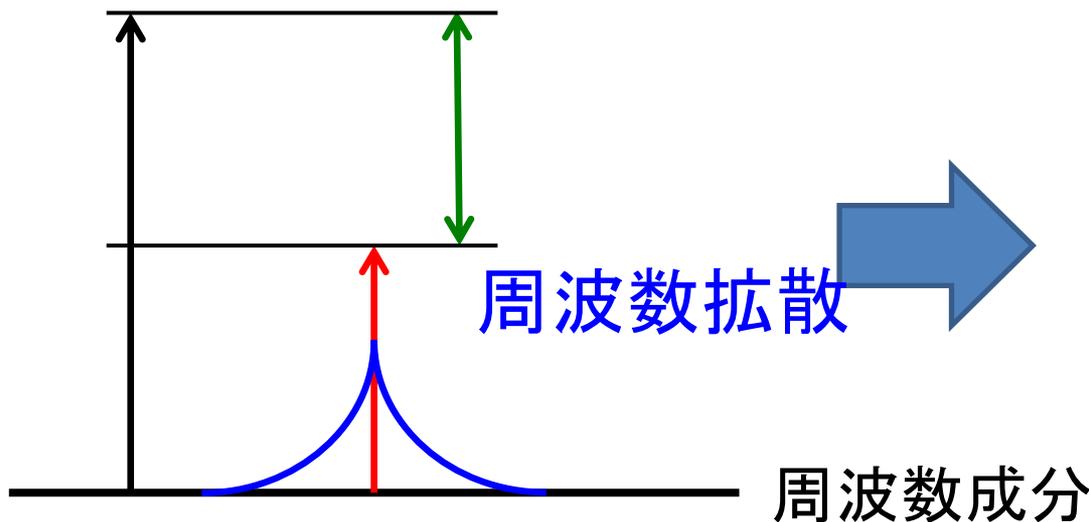
7I出力例
ランダム選択

$$\left\{ \begin{array}{l} I_{out} = 7I + \Delta I_4 + \Delta I_2 \\ I_{out} = 7I + \Delta I_4 + \Delta I_1 + \Delta I_0 \\ I_{out} = 7I + \Delta I_3 + \Delta I_2 + \Delta I_1 + \Delta I_0 \end{array} \right.$$

3通りの I_{out} の中から
ランダムに組み合わせを選択
歪み成分を周波数拡散させる



SFDR向上！



フィボナッチ数列を用いたDA変換器の高精度化アーキテクチャ

① 冗長性の利用

一つのデジタル入力に対して、
複数の「オンになる電流源の組」が存在

A. 最も線形になる「オンになる電流源の組」を選択
⇒静的線形性の向上

B. ランダムに「オンになる電流源の組」を選択
⇒SFDRの向上

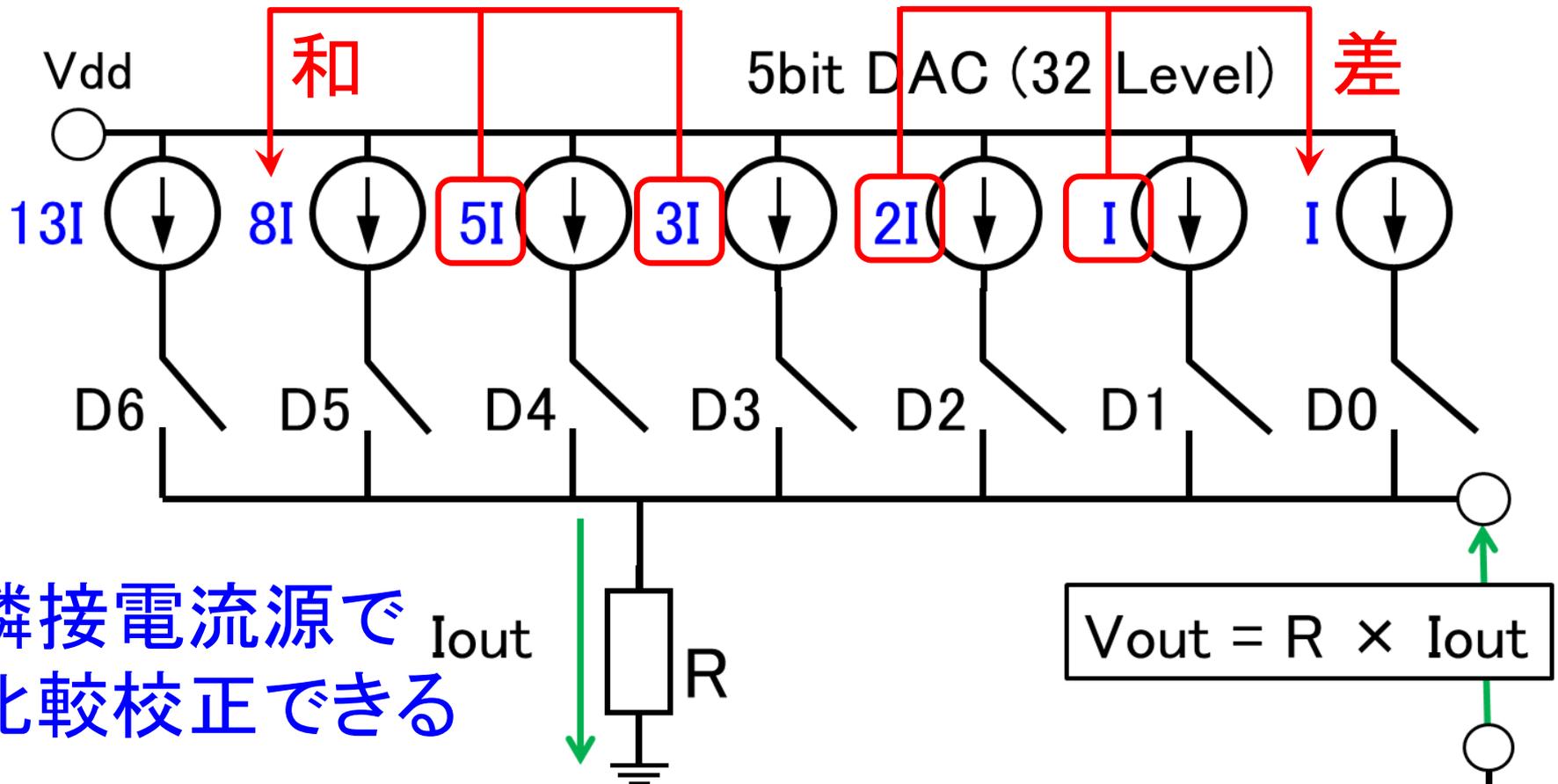
② $F_{n+2} = F_n + F_{n+1}$ の関係式利用

下位2つの電流源の和が次の電流源になることを利用
⇒電流源を比較・校正

DACの比較・校正

ECT-13-087

フィボナッチ数列の性質：近接項の和が次の項になる



- 研究背景、目的
- 冗長性を持つSAR ADCについて
- フィボナッチ数列を用いた冗長性設計
- 理論シミュレーション結果
- 電流源DACへの応用
- **まとめ、今後の課題**

まとめ

- SAR ADCの冗長設計理論の考察を行い
参照電圧重み付け法を提案した
- フィボナッチ数列を用いたDACの
冗長設計法及び自己校正法を検討した

今後

- フィボナッチ数列の新たな応用法を検討
- トリボナッチ数列(1.8進)への応用を検討

- シミュレーションの何パーセント補正できるというのは適切であるのか？どの値のどの誤りに対して出力を補正できるのかを明示する方が良いのではないか。
- エンコーダーの構成を考えているか？
- DACに関してトランジスタレベルで考えるとバイナリでもウナリでも電流源をカレントミラーとして使っているならば最終的なミスマッチは同じではないか？

- DACの隣接する電流源を用いて比較校正する方法(DACの3つ目の応用方法)は実際の使用方法を検討してあるのか？
- SAR ADCの場合、比較電圧を発生させるDACの精度が問題となるが、この補正を使うことでそちらの条件を緩和することができるか？