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

群馬大学　電子情報数理教育プログラム
小林研究室　修士1年
小林　佑太朗

Kobayashi Laboratory

Supported by STARC

群馬大学 GUNMA UNIVERSITY

Outline

- 研究目的と概要
- SAR ADCの冗長設計
- 補正能力の一般化と従来設計手法
- フィボナッチ数列を用いた冗長設計SAR ADC
- 整数論を用いた冗長設計SAR ADC
  - Nーボナッチ数列
  - リュカ数列
- まとめ

Outline

- 研究目的と概要
- SAR ADCの冗長設計
- 補正能力の一般化と従来設計手法
- フィボナッチ数列を用いた冗長設計SAR ADC
- 整数論を用いた冗長設計SAR ADC
  - Nーボナッチ数列
  - リュカ数列
- まとめ

研究背景と目的

自動車に付加価値や競争力をつける
カーエレクトロニクス技術に注目

↓ 著しい発展

車載システム中のマイコンと組み合わせる
“逐次比較近似AD変換器”への性能要求が厳しく ☹

↓

逐次比較近似AD変換器の冗長設計
（**高信頼性化・高速化**）

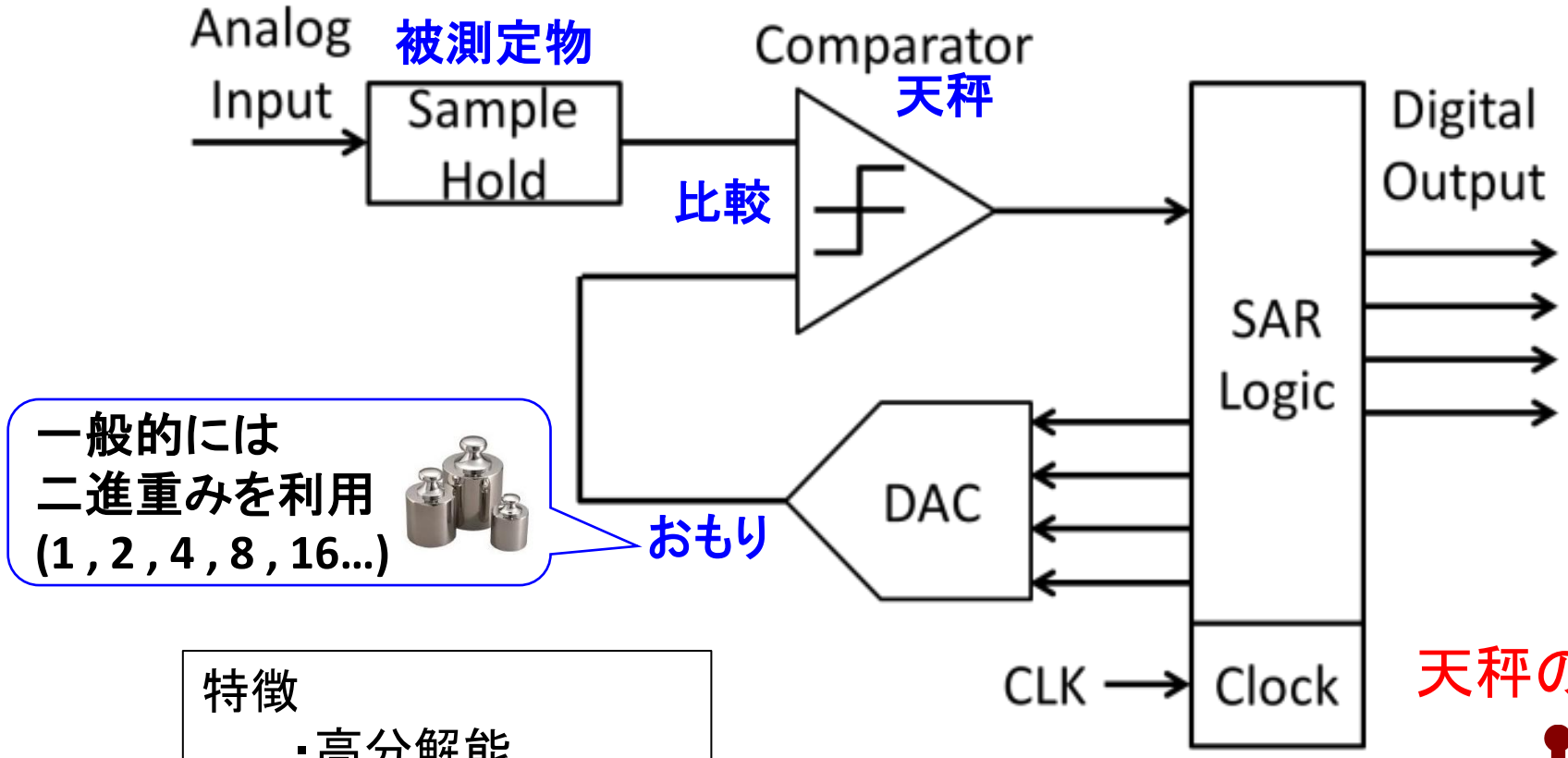
効率の良い設計方法？

研究目的

冗長設計ADC ＋ 整数論 ⇒ 高性能AD変換器

逐次比較近似ADC（SAR ADC）

アナログ入力と参照電圧を比較、結果に応じたディジタル出力



一般的には
二進重みを利用
(1 , 2 , 4 , 8 , 16…)

特徴
- 高分解能
- 中速サンプリング
- 小チップ面積
- 低消費電力

天秤の原理

二進探索SAR ADCの動作

5bit5step SAR ADC 変換動作

アナログ入力 Vin (被測定物) 7.3

比較電圧重みp(k)＝分銅

比較電圧重み **p(1)** 16

比較電圧重み **p(2)** 8

比較電圧重み **p(3)** 4

比較電圧重み **p(4)** 2

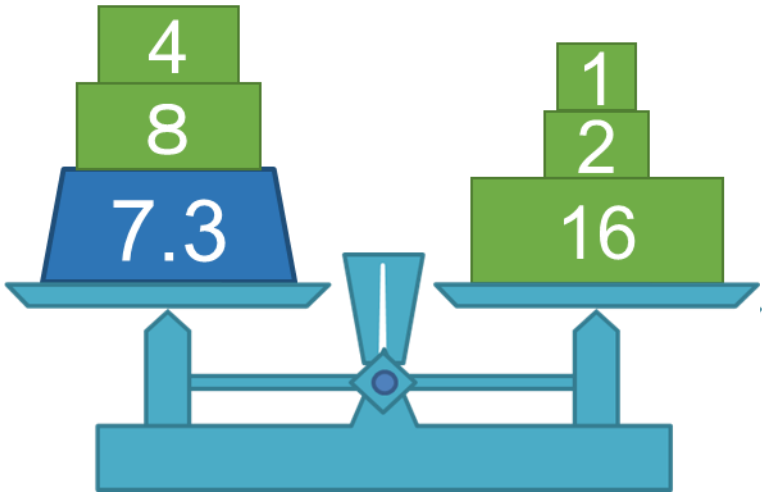
比較電圧重み **p(5)** 1

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						2
	1						1
	0						0

二進探索SAR ADCの動作

5bit5step SAR ADC 変換動作

比較（重み：16, 8, 4, 2, 1）



7.3 ⇒ 7

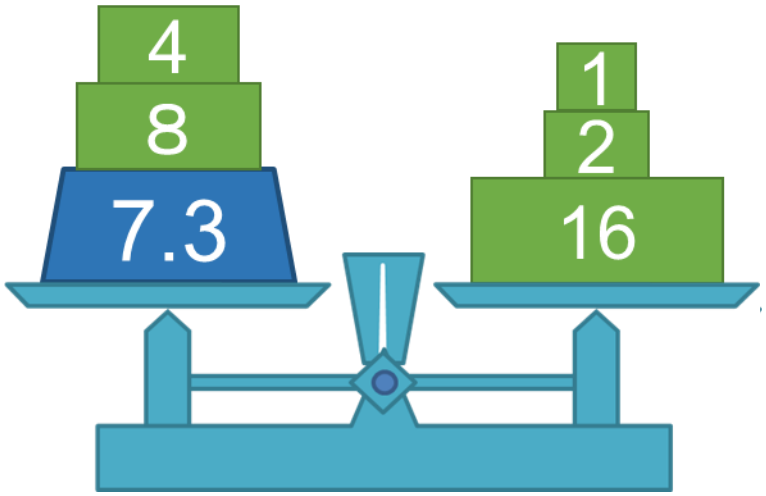
比較結果が…
右が大きい⇒0(重い)
左が大きい⇒1(軽い)

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

二進探索SAR ADCの動作

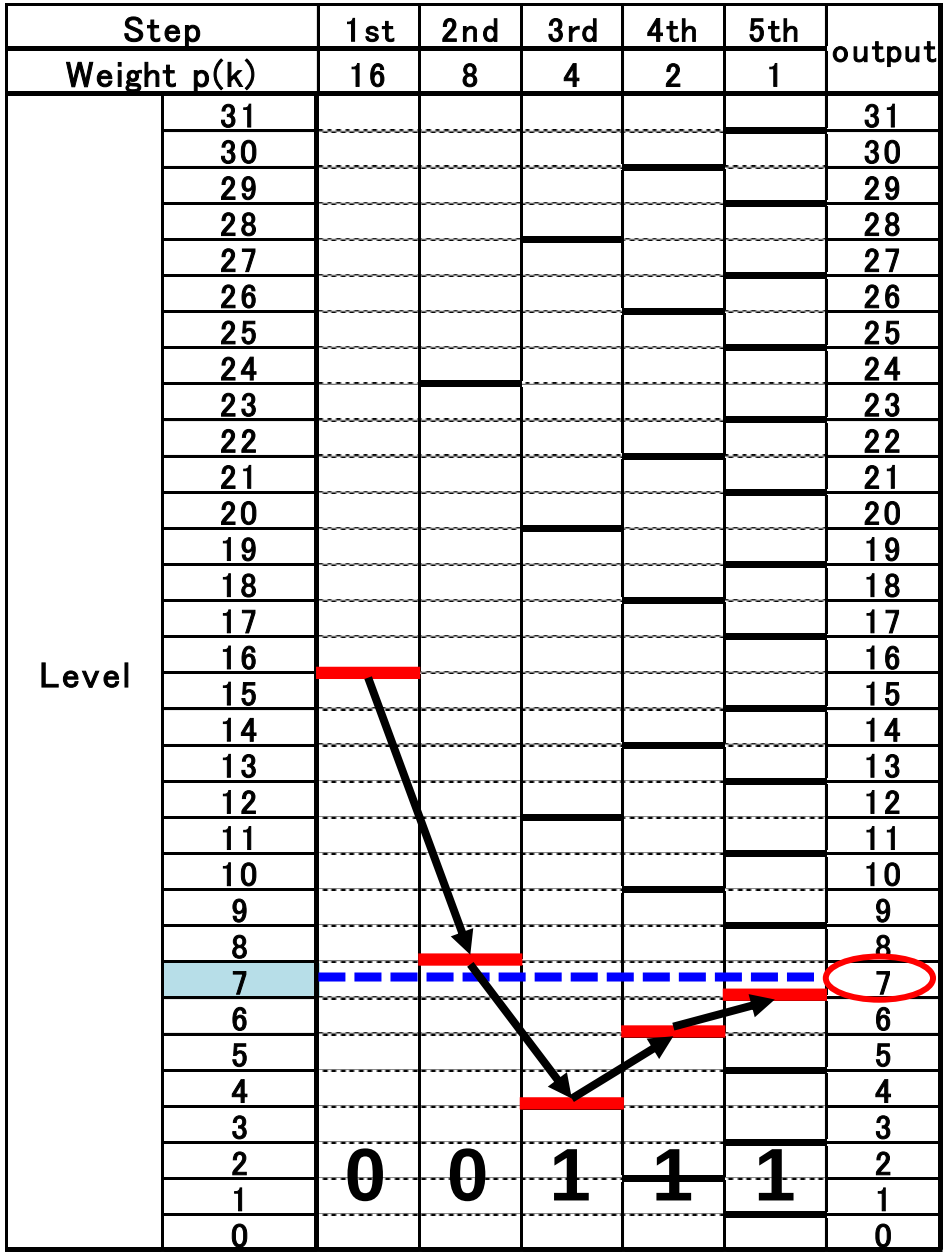
5bit5step SAR ADC 変換動作

比較（重み：16, 8, 4, 2, 1）



7.3 ⇒ 7

次の分銅を…
0 :左に載せる(マイナス)
1 :右に載せる(プラス)



二進探索SAR ADCの動作

SAR ADC変換例

二進重み利用

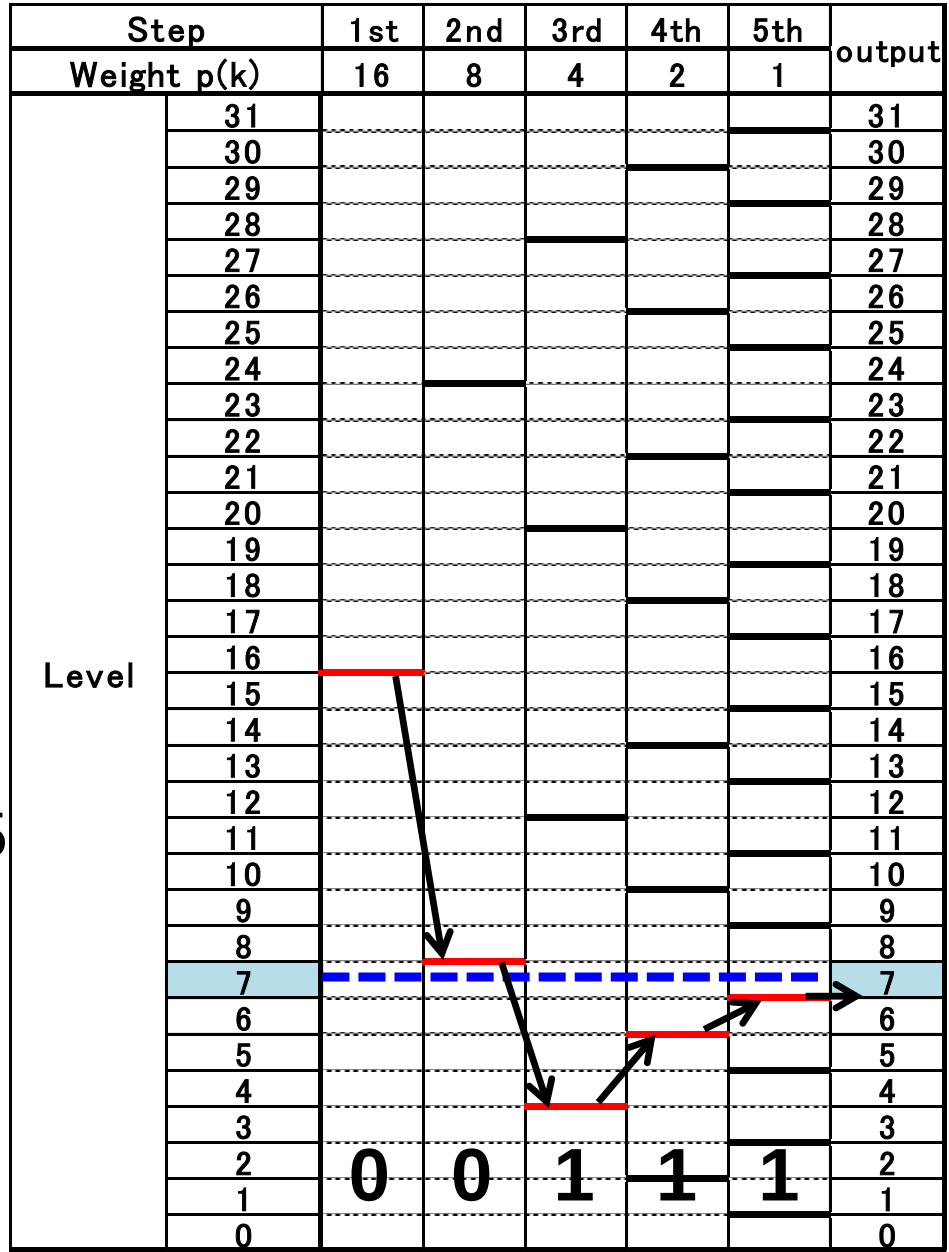
5bit5step　ADC

input 7.3 [LSB]

十進数と二進数が一対一に対応

7⇒$(00111)_2$

D_{out} =16 − 8 − 4 + 2 + 1 + 0.5-0.5



二進探索SAR ADCの動作

SAR ADC変換例

二進重み利用

5bit5step　ADC

input 7.3 [LSB]

十進数と二進数が一対一に対応

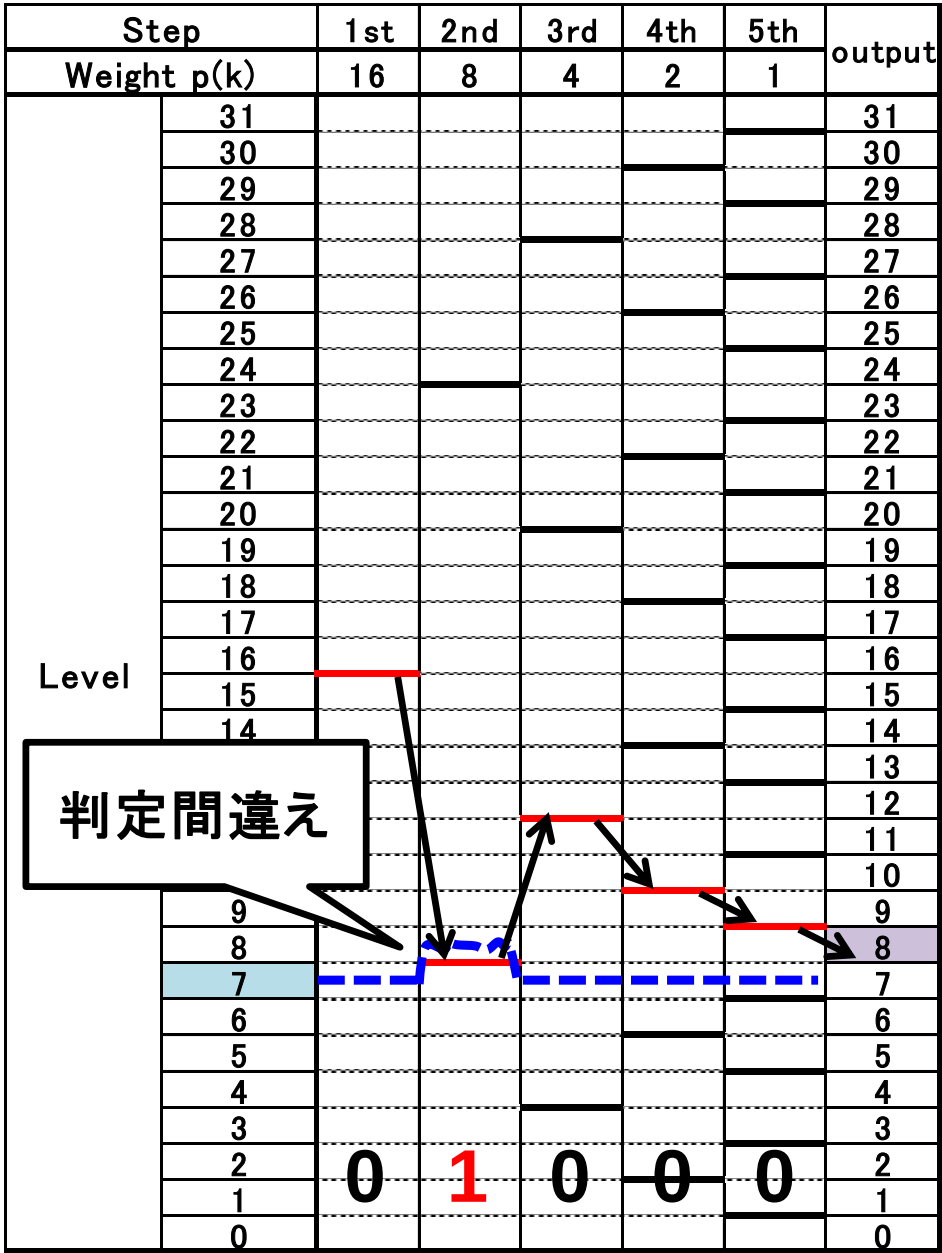
$7 \Rightarrow (00111)_2$

一回の判定誤りが
出力間違いの原因になる

$7 \Rightarrow (01000)_2 \Rightarrow 8$

信頼性の劣化

冗長設計で改善



Outline

- 研究目的と概要
- SAR ADCの冗長設計
- 補正能力の一般化と従来設計手法
- フィボナッチ数列を用いた冗長設計SAR ADC
- 整数論を用いた冗長設計SAR ADC
  - Nーボナッチ数列
  - リュカ数列
- まとめ

冗長性を持つSAR ADC

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

二進重み　1,2,4,8,16

↓

非二進重み　1,2,3,6,10,16

↓ 5step ⇒ 6step など

ディジタルコードによる表現方法増加

↳ SAR ADC
- 誤り耐性向上
- 変換速度向上

冗長探索SAR ADCの動作

SAR ADC変換例

冗長重み利用

5bit6step ADC

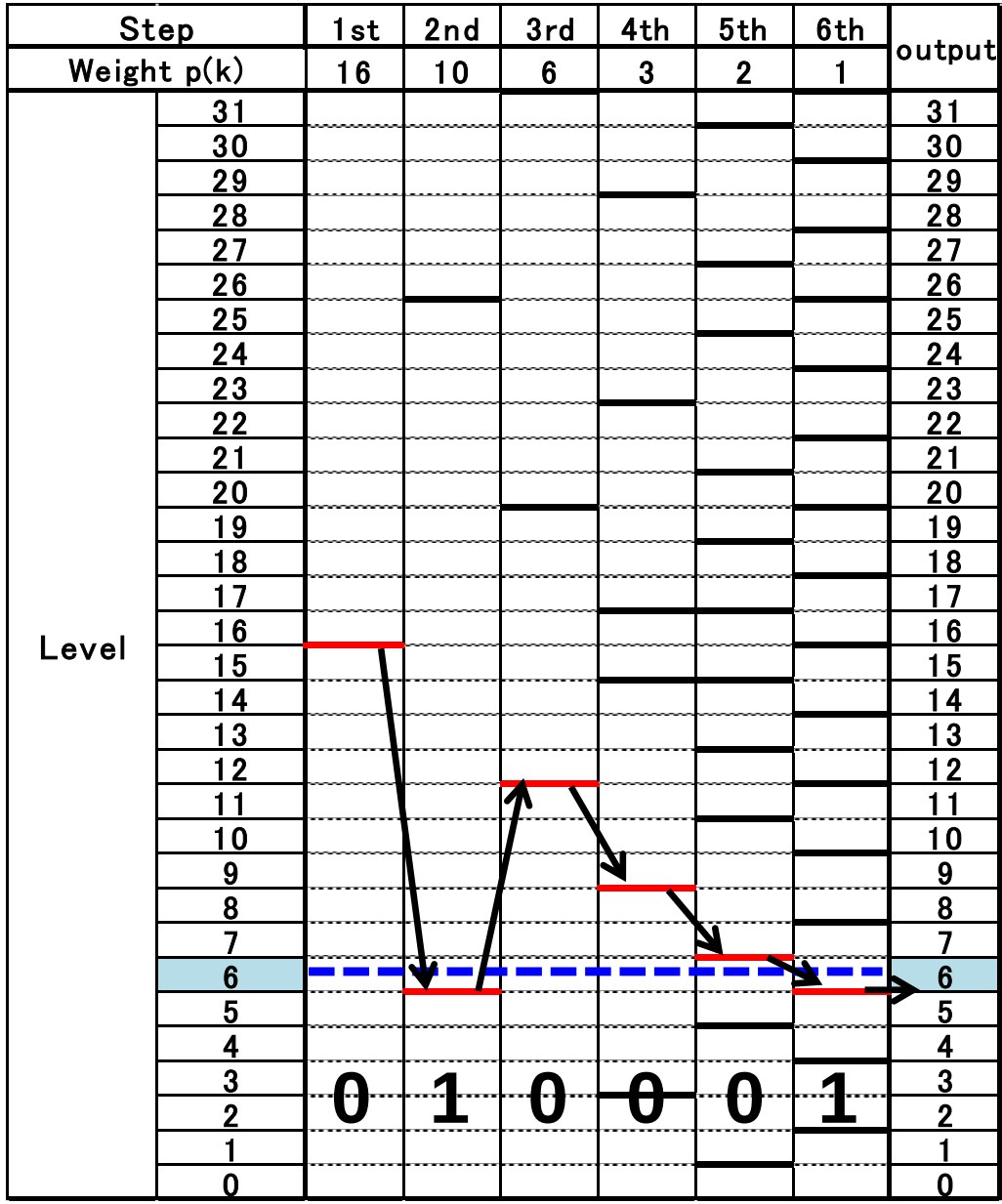
input 6.3 [LSB]

判定ステップ増加

6⇒010001

$D_{out} =$

16 − 10 + 6 − 3 − 2 − 1 + 0.5 − 0.5



冗長探索SAR ADCの動作

SAR ADC変換例

冗長重み利用

5bit6step ADC

input 6.3 [LSB]

判定ステップ増加

6⇒010001

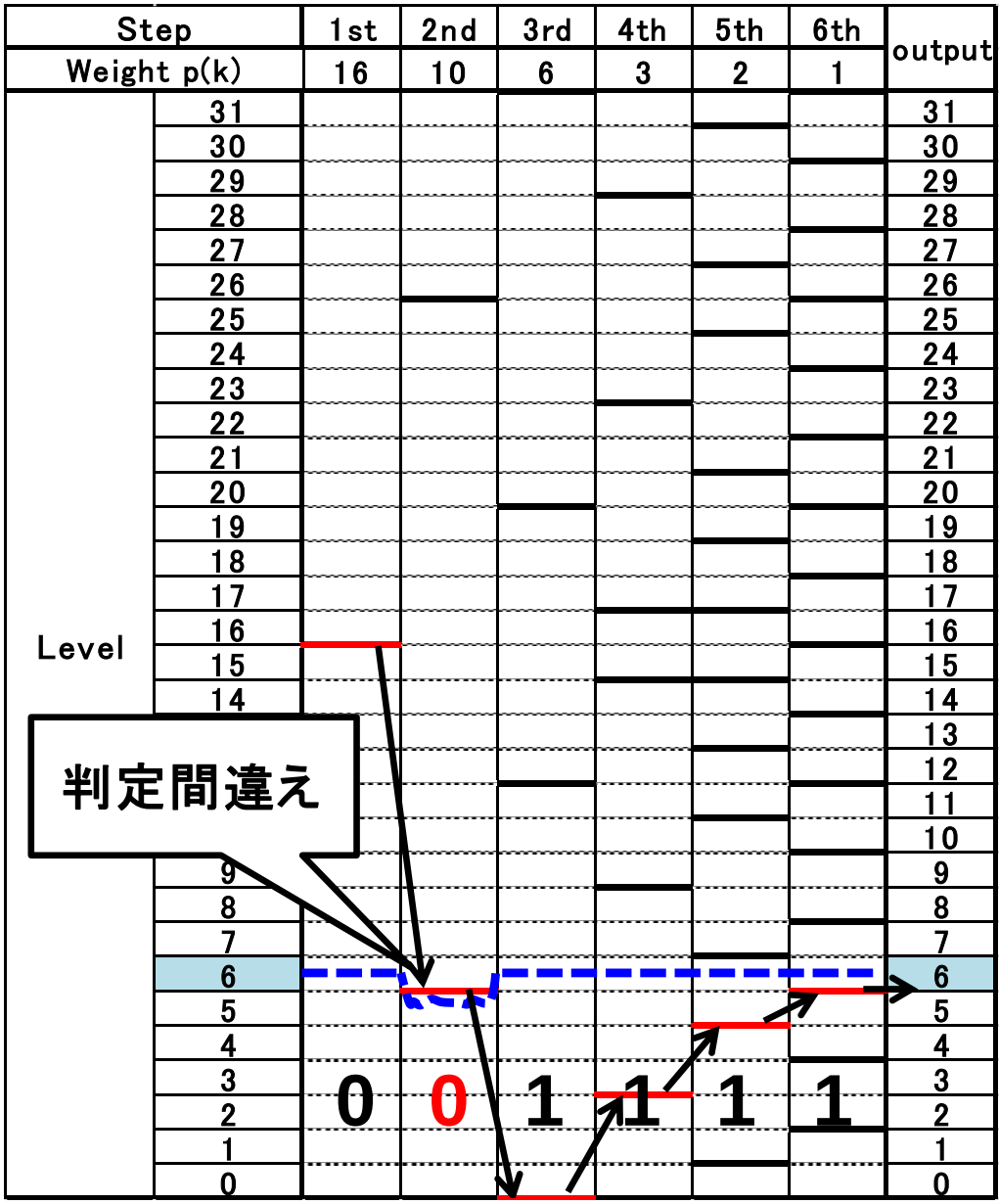
ディジタルコード表現が複数存在

6⇒001111⇒6

後段の判定誤りを補正

ディジタル誤差補正実現

誤り耐性向上



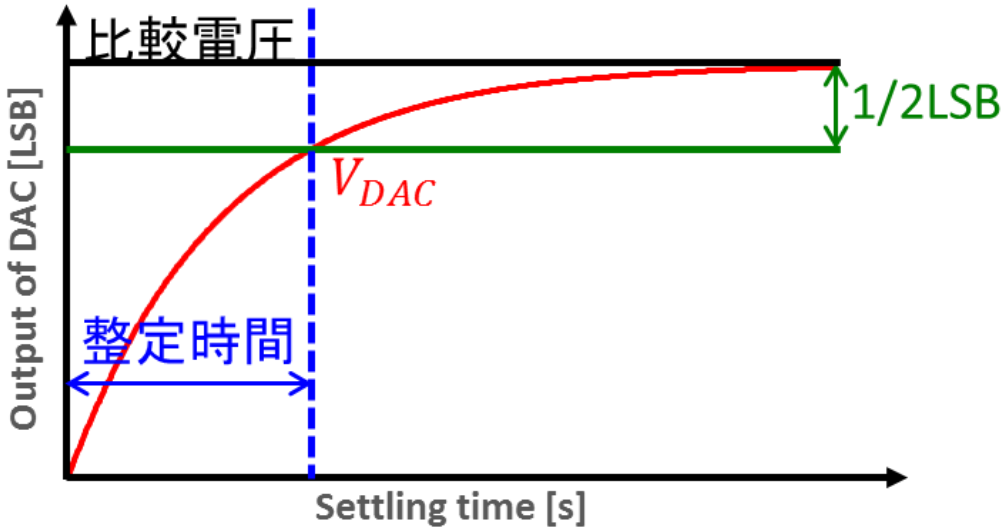
冗長設計による速度向上

二進探索SAR ADC
DACの出力⇒完全整定

変換時間の増長

非二進探索SAR ADC

後段で補正可能



5bit SAR ADC

二進探索(完全整定)

Step1	Step2	Step3	Step4	step5

AD変換時間

非二進探索(不完全整定)

Step1	Step2	Step3	Step4	Step5	Step6	Step7

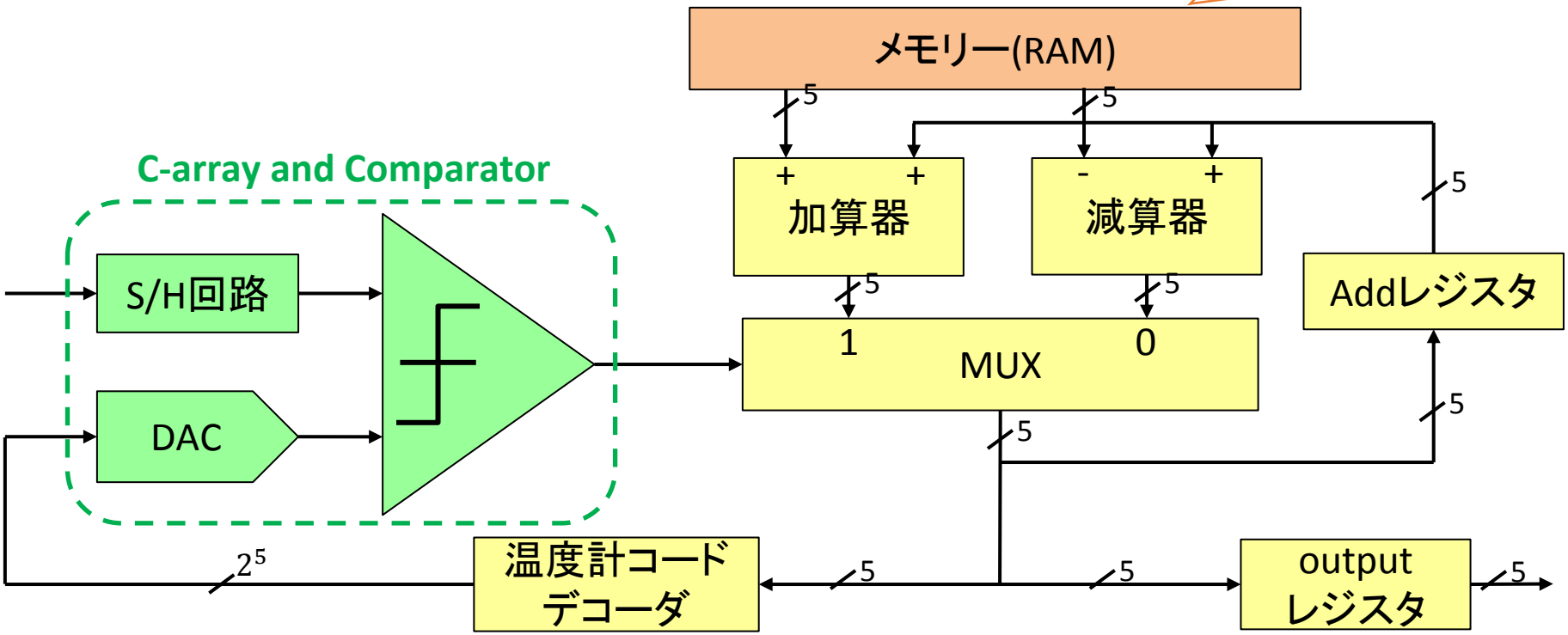
不完全整定誤差補正

AD変換時間

DACの不完全整定許容⇒変換速度向上を実現

冗長SAR ADCの実現

5bit　冗長SAR ADC



付加回路は小面積なディジタル回路

冗長設計　⇒　誤り耐性 and/or 変換速度に優れたAD変換器

Outline

- 研究目的と概要
- SAR ADCの冗長設計
- 補正能力の一般化と従来設計手法
- フィボナッチ数列を用いた冗長設計SAR ADC
- 整数論を用いた冗長設計SAR ADC
  - Nーボナッチ数列
  - リュカ数列
- まとめ

補正できる入力範囲差q

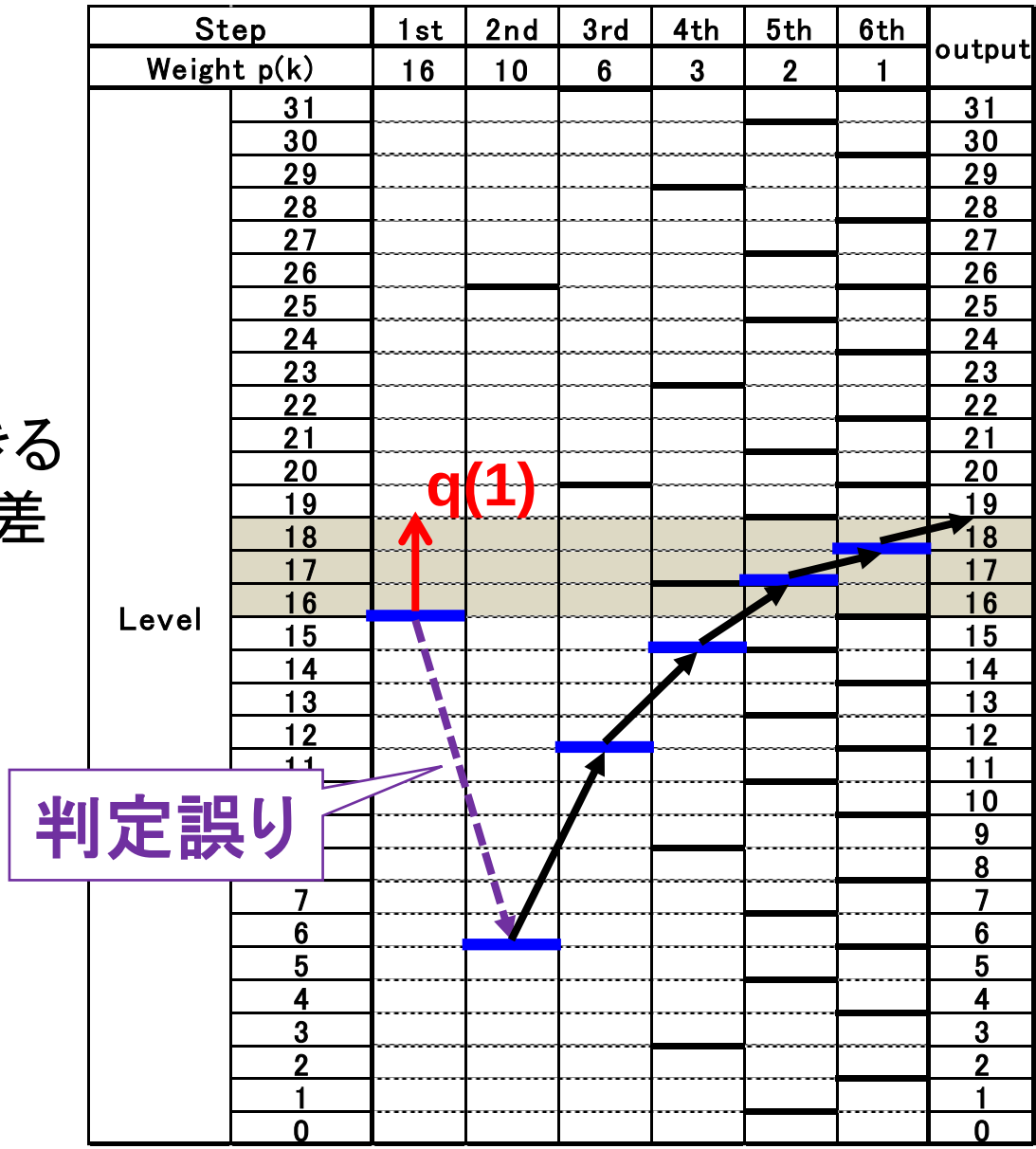
SAR ADC変換例

冗長重み利用

5bit6step　ADC

補正能力q(k)の定義

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

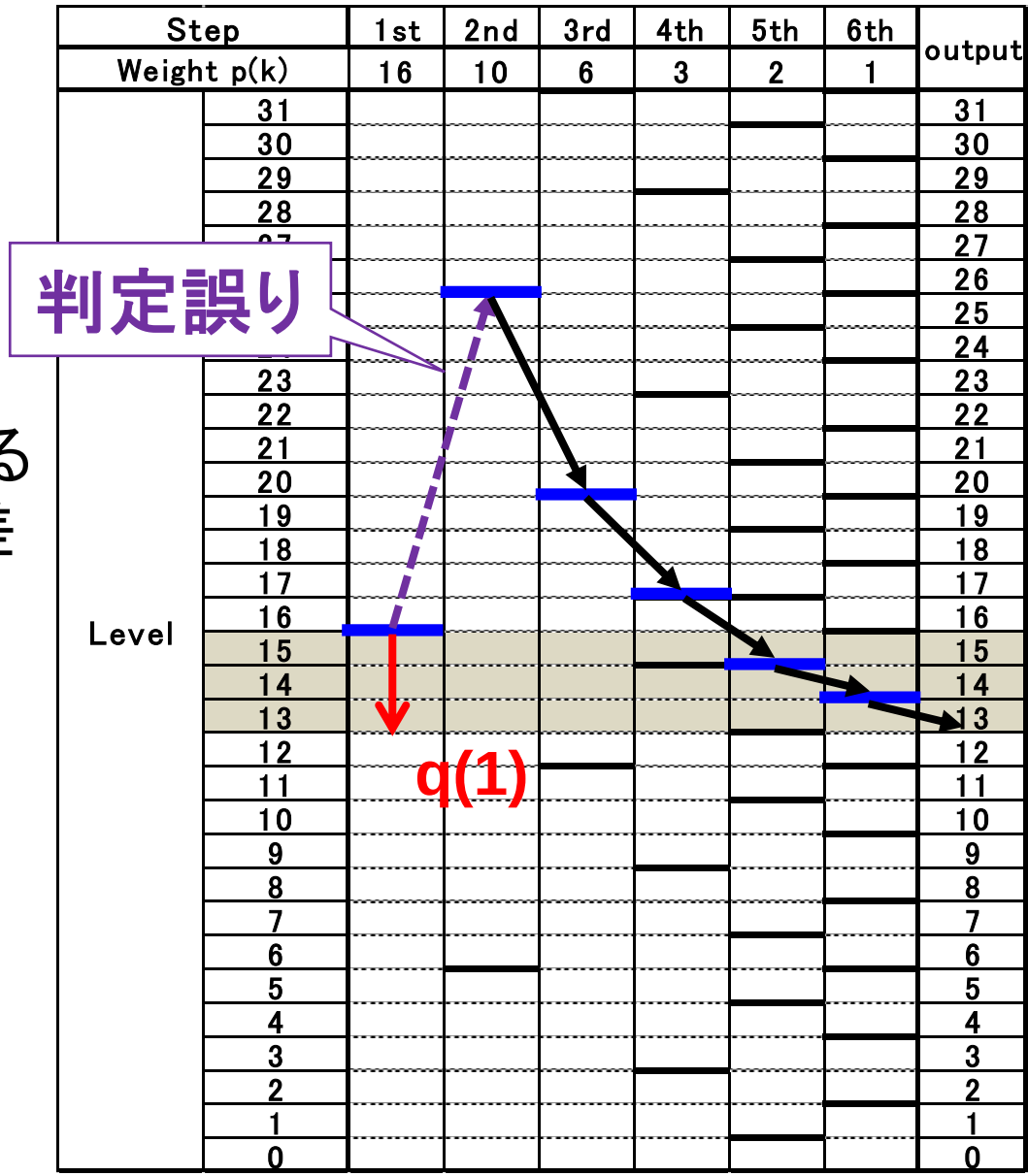


補正できる入力範囲差q

SAR ADC変換例
冗長重み利用
5bit6step　ADC

補正能力q(k)の定義
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

比較電圧に対して対称



補正できる入力範囲差q

SAR ADC変換例

冗長重み利用

5bit6step　ADC

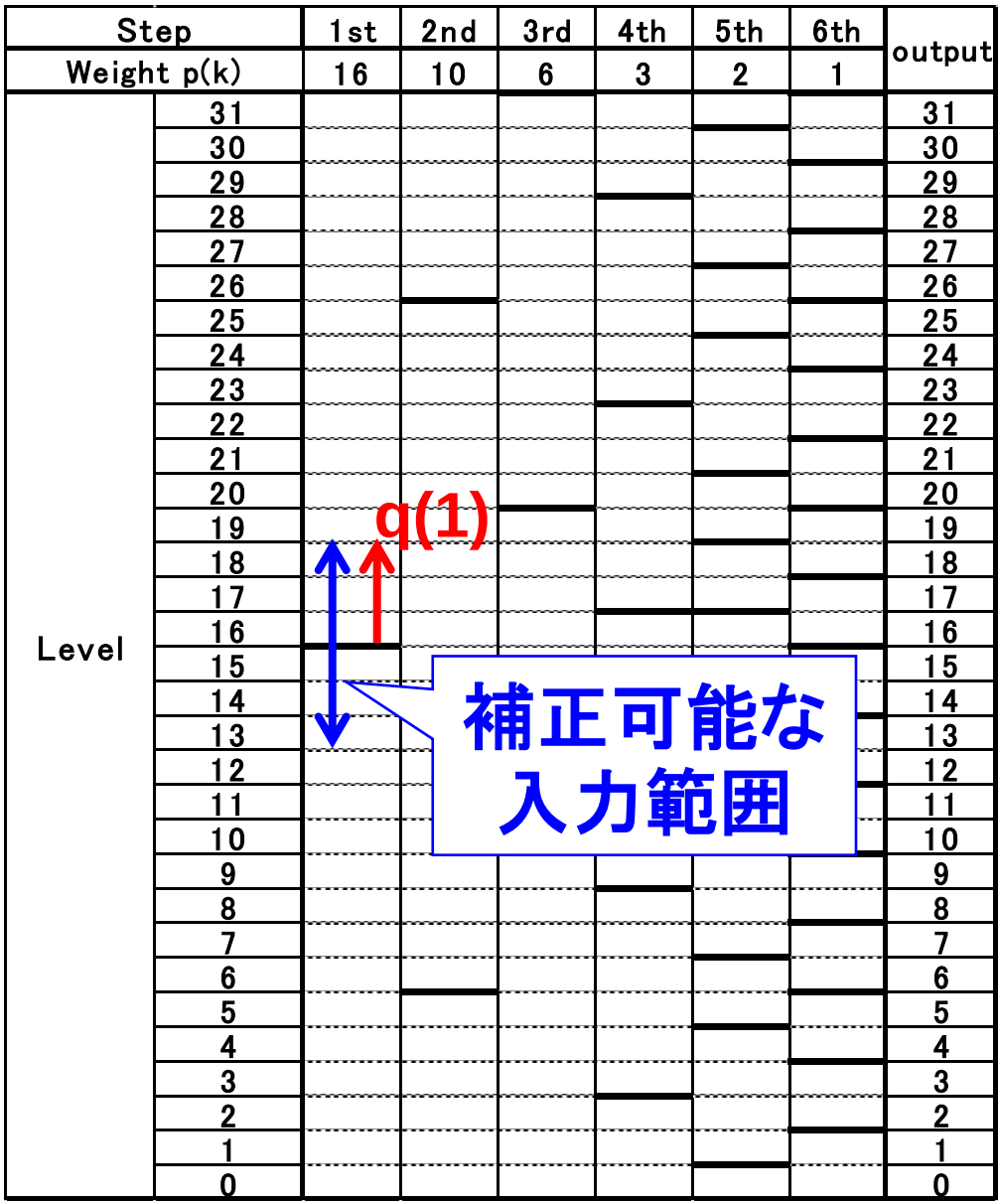
補正能力q(k)の定義

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

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

q(k)が大きい＝補正力高い



補正できる入力範囲差q

SAR ADC変換例
冗長重み利用
5bit6step　ADC

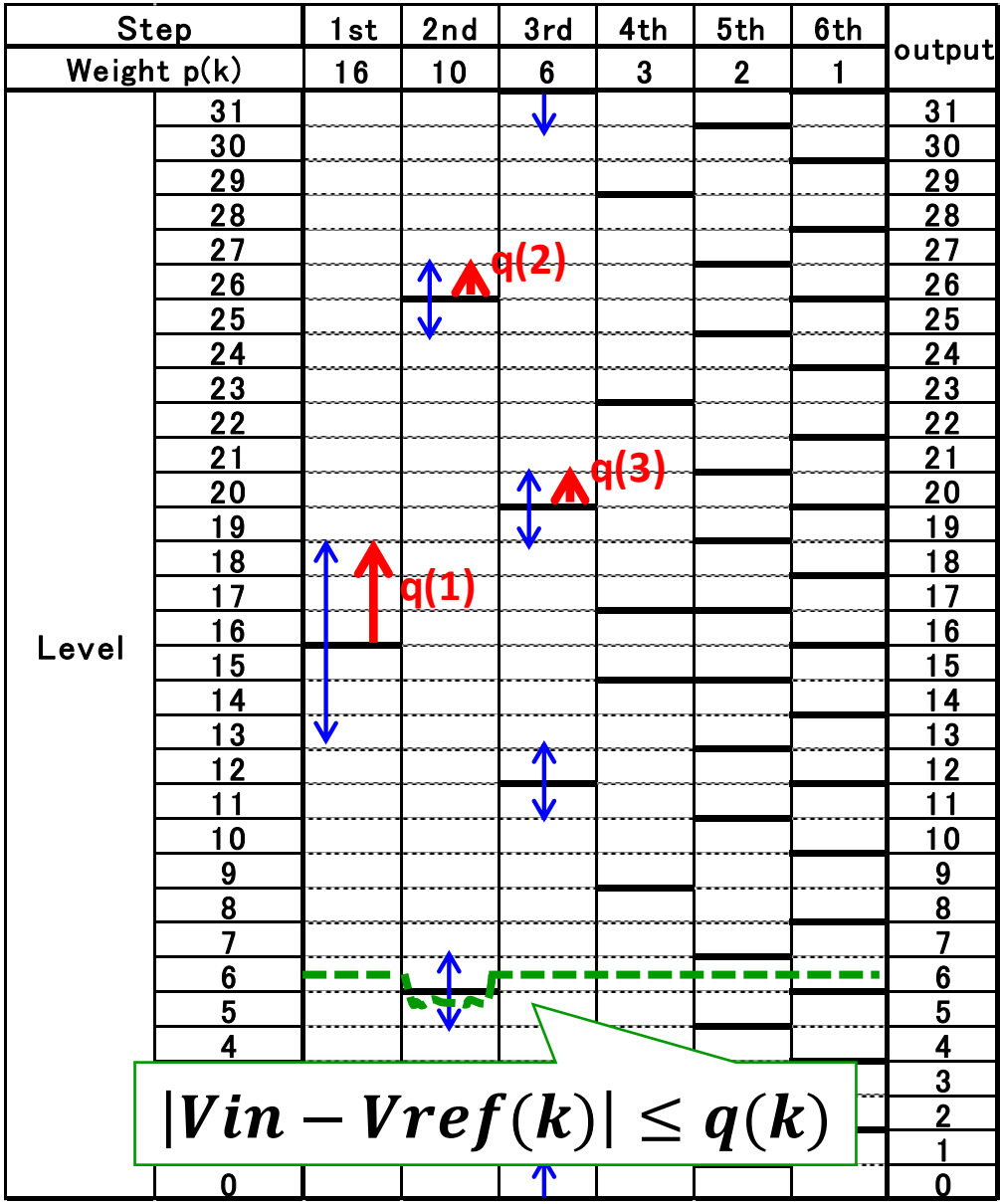
k step目で判定誤りがあっても
後段で補正が可能な入力範囲差q(k)は

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

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？

？



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　（ただし　$p(1) = 2^{N-1}$）

従来手法

①基数radixから決定する　⇒　$p(k) = r^{M-k}$ (ただし$1 < r < 2$)

- 適切な基数の決定が難しい
  - 補正力と変換ステップ数はトレードオフ

- p(k)は必ず小数になる(単位項による実現困難)
  - 面積比を基数比で利用する　⇒　精度が悪い
  - 四捨五入して整数を使う　　⇒　step毎のq(k)の大小ばらつき

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させる

従来手法の問題点

5bit6step　ADC

冗長設計手法①

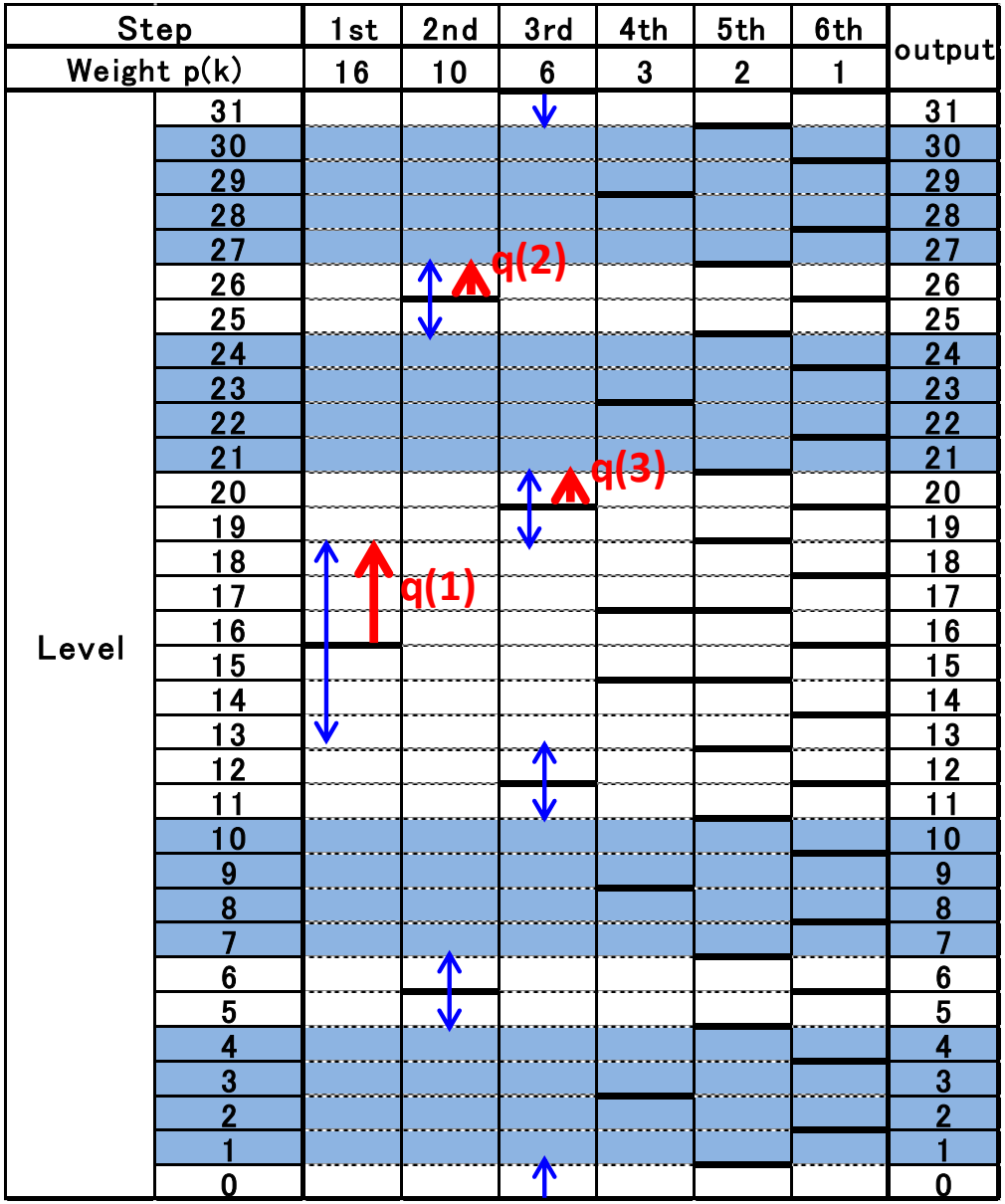
radix=1.80

比較電圧重みp(k)

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

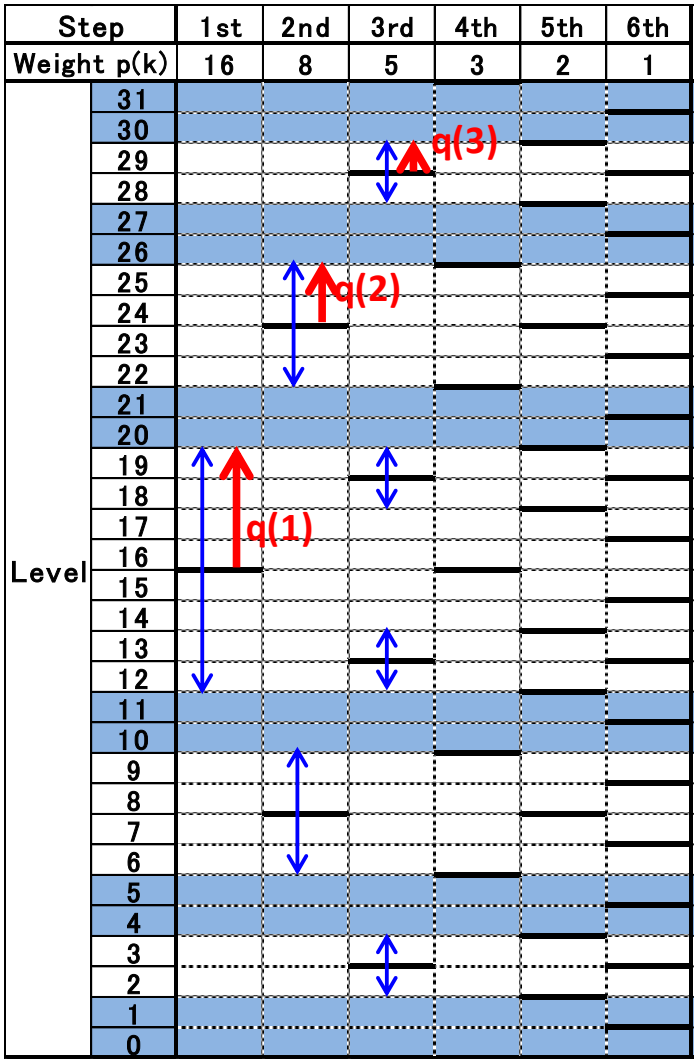
原理的に補正不可能な
入力範囲が存在
冗長設計効果の劣化

適切なp(k)選択手法が重要

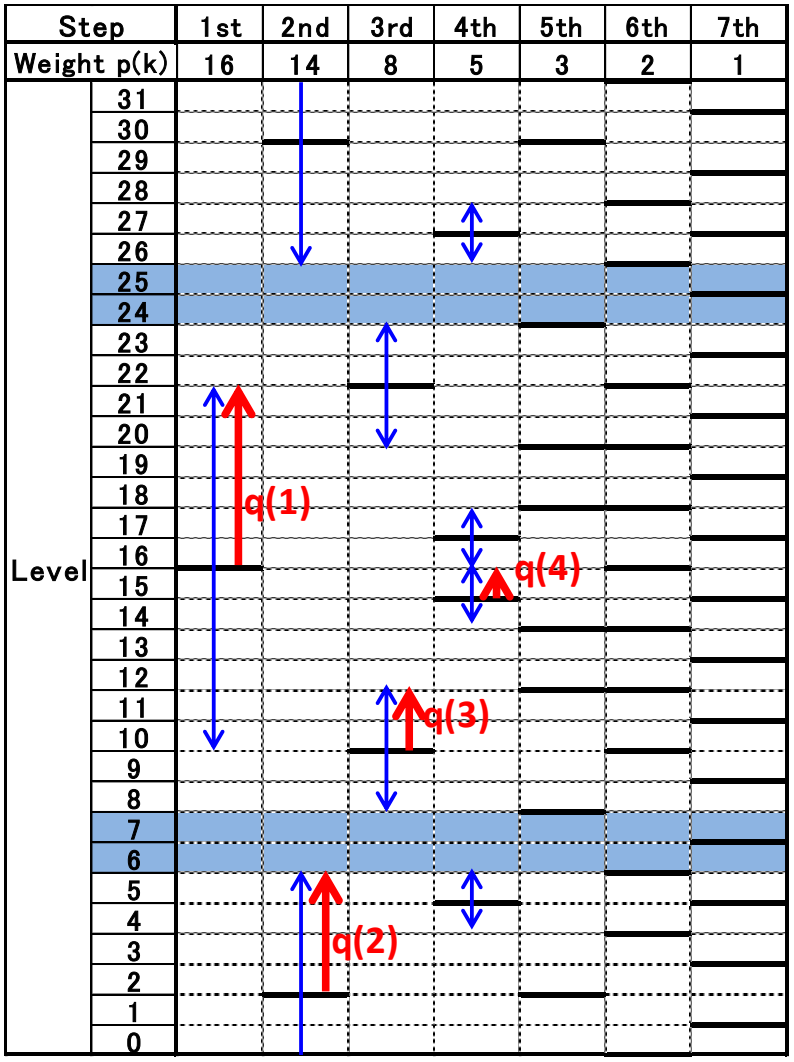


従来手法の問題点

5bit6step (radix=1.7)



5bit7step (radix=1.7)



ステップ数を増加させてもすべての範囲をカバーできない ⇒ 適切な基数の決定

Outline

- 研究目的と概要
- SAR ADCの冗長設計
- 補正能力の一般化と従来設計手法
- フィボナッチ数列を用いた冗長設計SAR ADC
- 整数論を用いた冗長設計SAR ADC
  - Nーボナッチ数列
  - リュカ数列
- まとめ

フィボナッチ数列とは

フィボナッチ数列

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

Leonardo Fibonacci
(伊：1170-1250年頃)

初めの項を計算すると

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

また隣り合う項の比率は以下の値φに収束する

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

収束比率 φ：黄金比（Golden ratio）

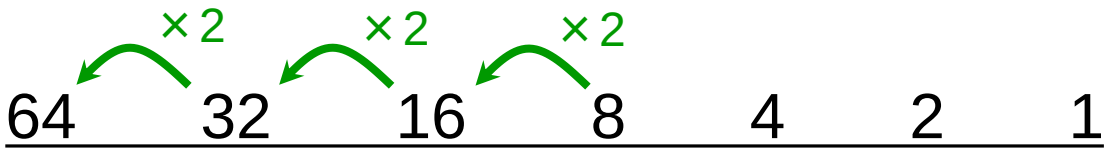
比較電圧重みp(k)の決定（提案手法）

$N\ bit$ 全$M\ step$中　$k\ step$目の比較重み$p(k)$を決定　（ただし　$p(1) = 2^{N-1}$）

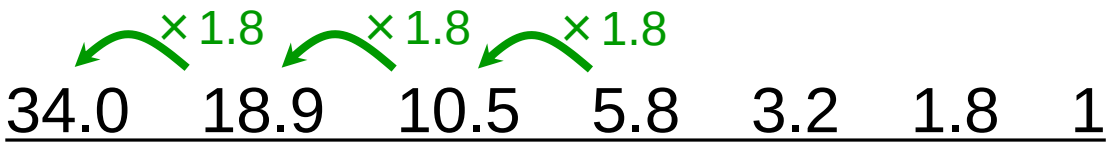
提案手法

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

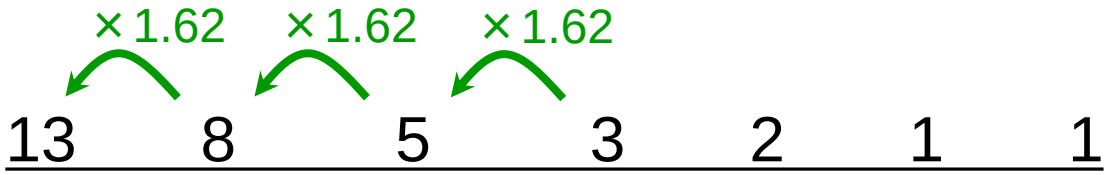
Binary Weight
二進数



Radix1.8 Weight
1.8進数



Fibonacci Weight
約1.62進数



隣り合う項の比率が黄金比φに収束する性質

⇒整数のみで約1.62進数$(radix = 1.62)$を実現できる！

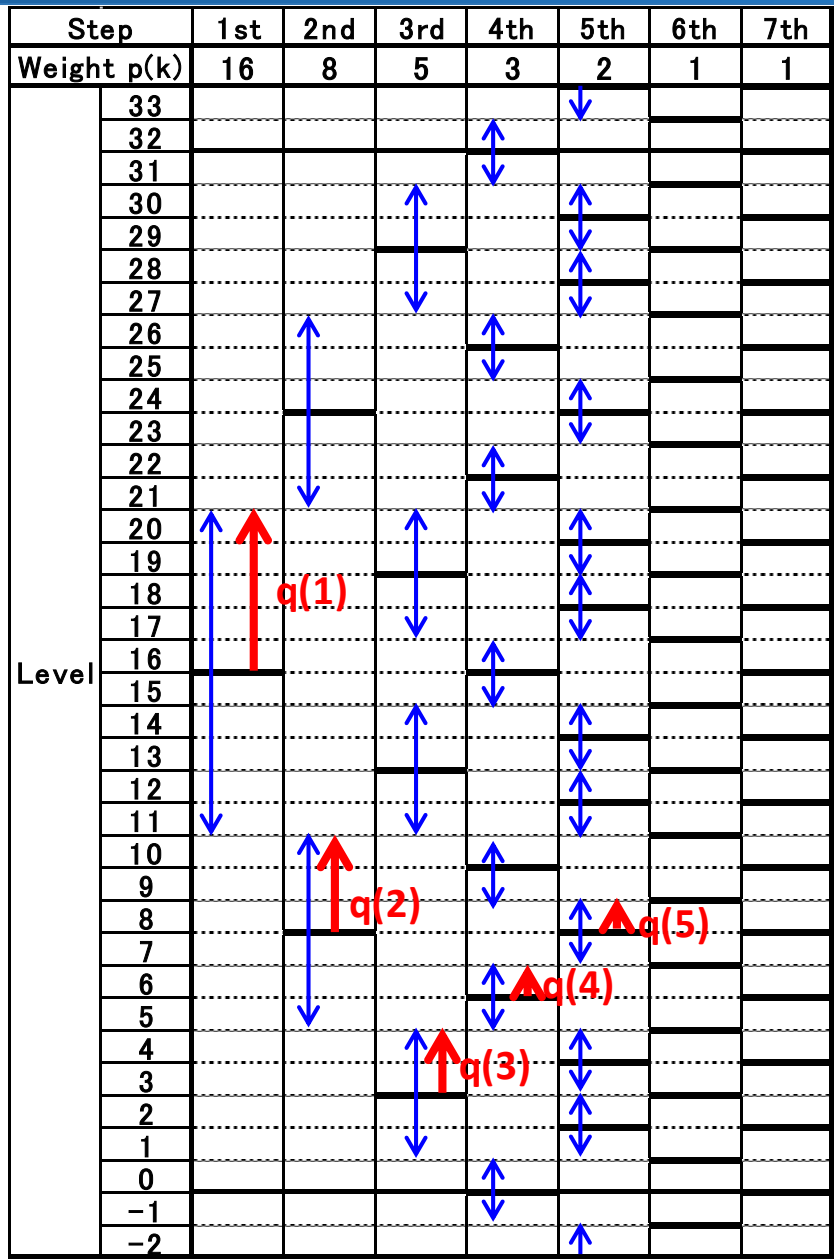
フィボナッチ数列を用いたSAR ADC

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



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

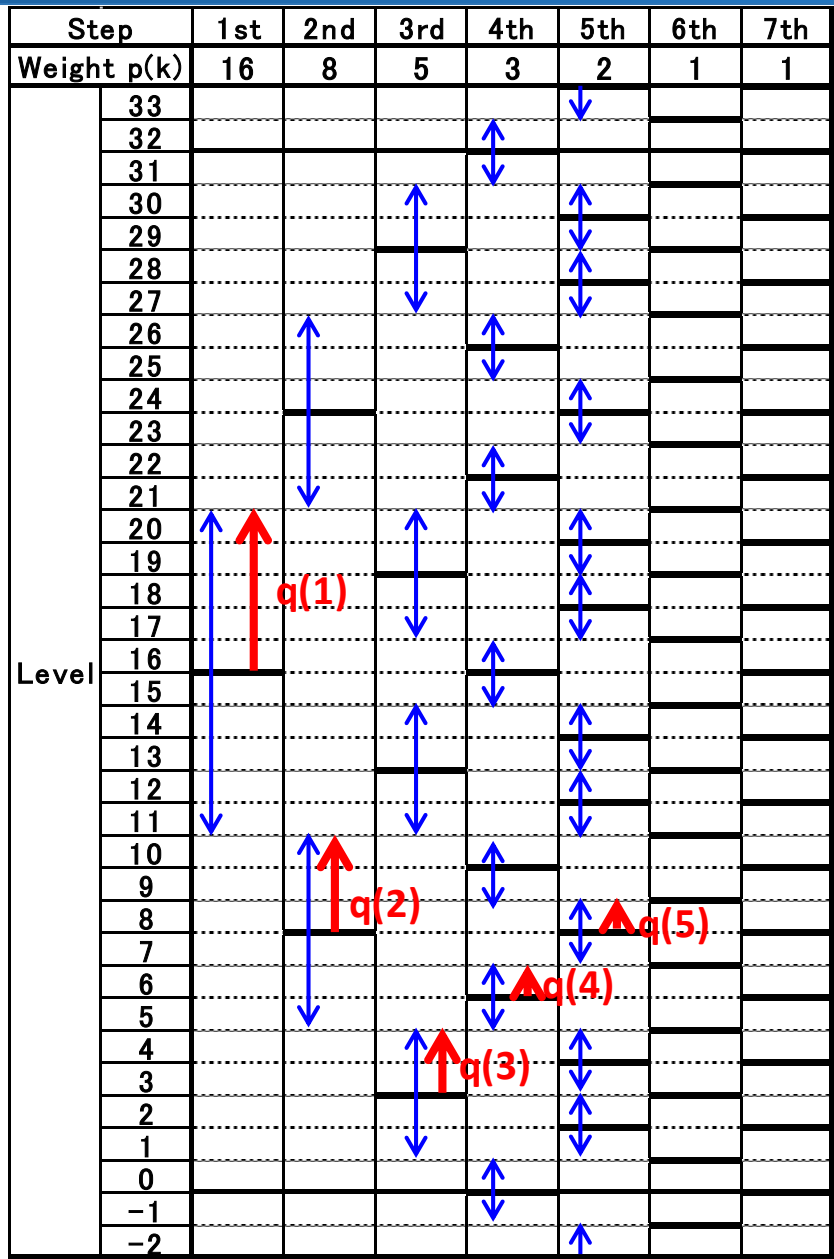
②許容できる範囲が必ず接する

性質①

許容値q(k)は必ずフィボナッチ数であり
その値はF_{M-k-1}となる

$F_0 = 0$を最小値とすれば、これはすなわち
最後から2stepは必ず誤差許容値が0であ
ることに等しい

論文中の証明参照



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

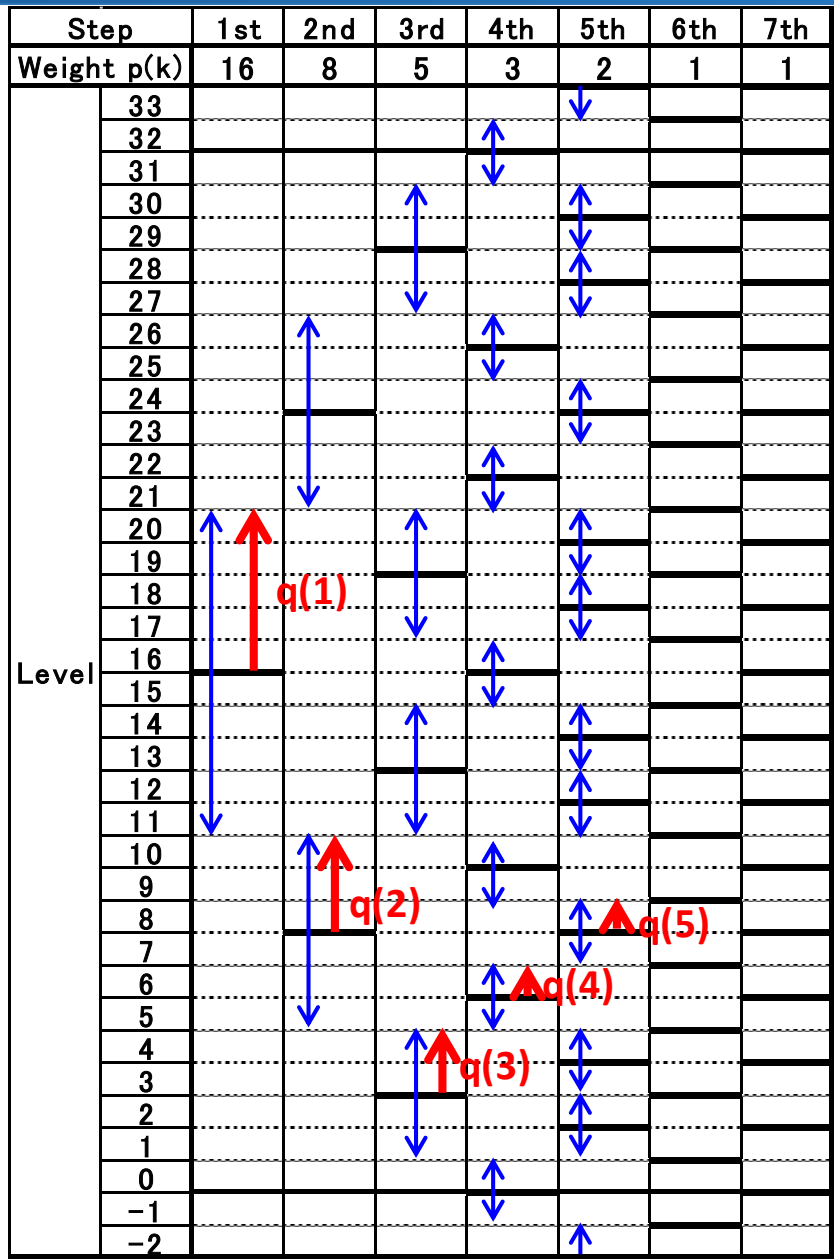
②許容できる範囲が必ず接する

性質②

k step目の補正可能範囲はk+1 step目の補正可能範囲と重なることなく必ず接する
(k stepとk+1 stepの両矢印の先端は必ず同じLevel値となる)

これは同時に
補正可能範囲が重なる／離れるの境界がフィボナッチ数重みであること、
フィボナッチ数重みが補正可能範囲を接させるための最速の重み付けであることを示す

論文中の証明参照



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！

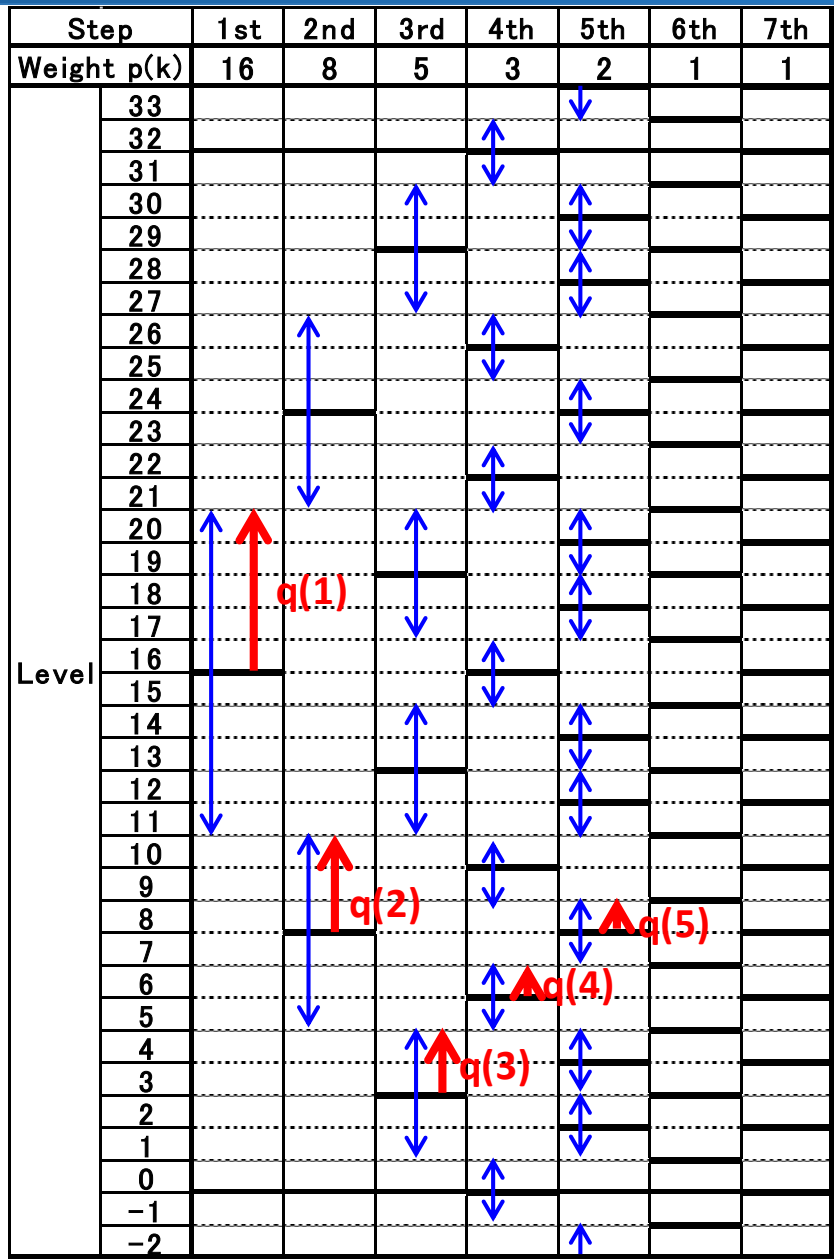
①許容値q(k)は必ずフィボナッチ数

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 ⇒ q(k)は離れる
- 基数 rが小さい ⇒ q(k)は重なる

黄金比φは
冗長度の境界条件であり
設計指針となる



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

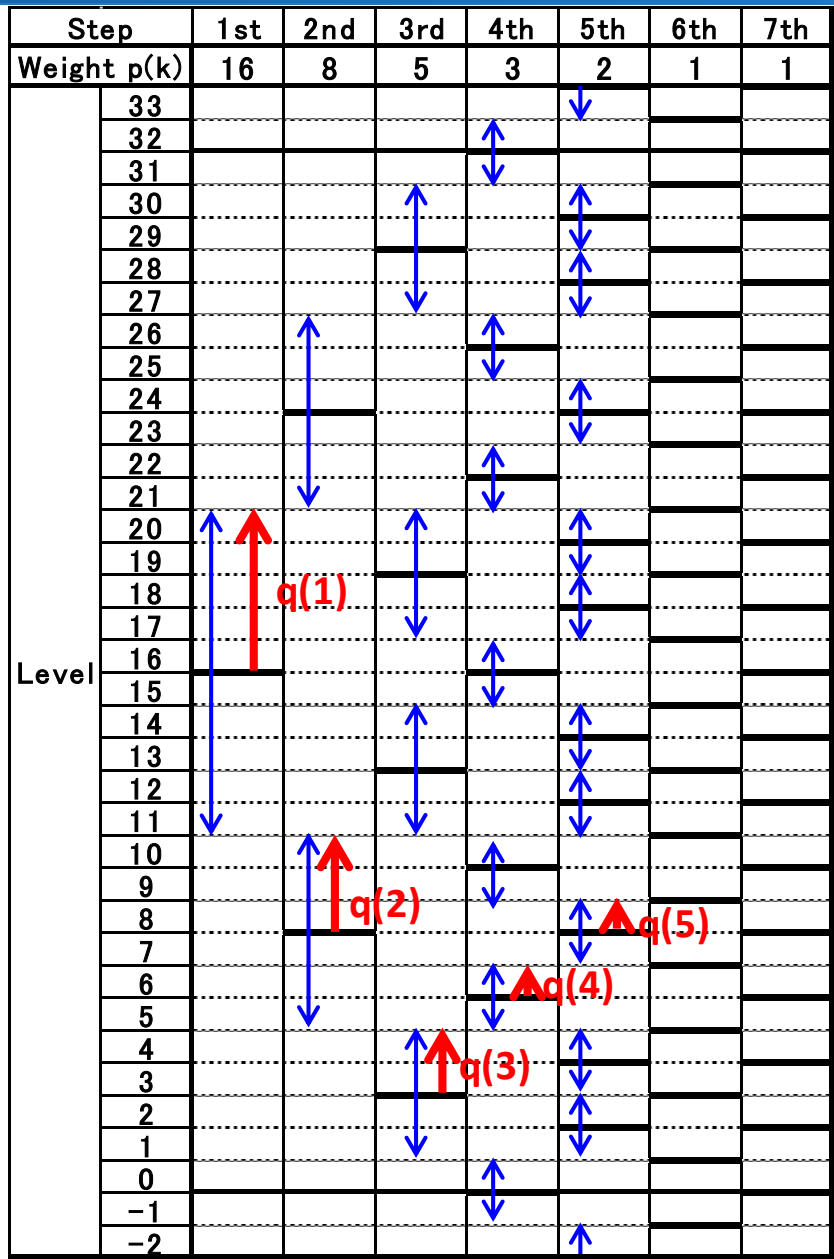
↓

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φを使うことで
- 無駄なステップ
- 補正できない入力範囲

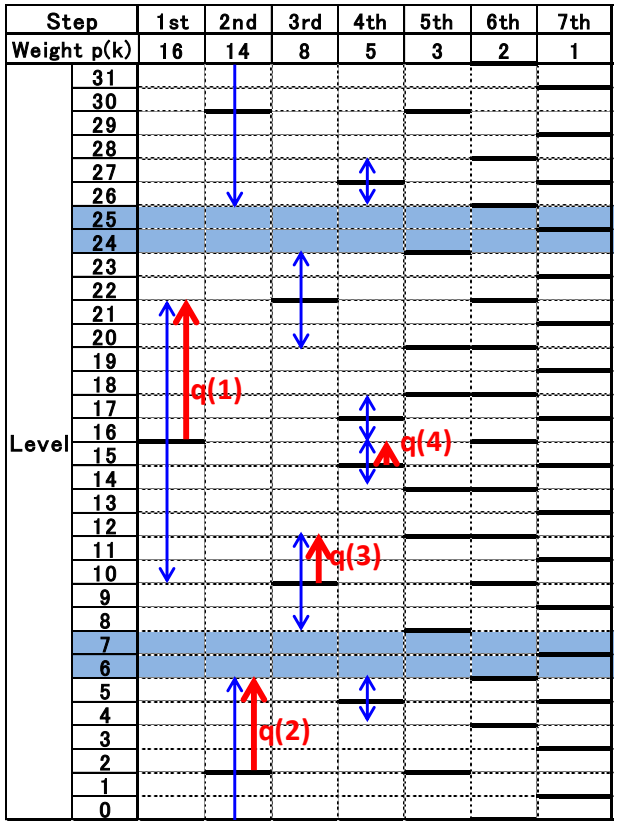
がない最も効率のよい設計が可能



従来手法との比較(5bit ADC)

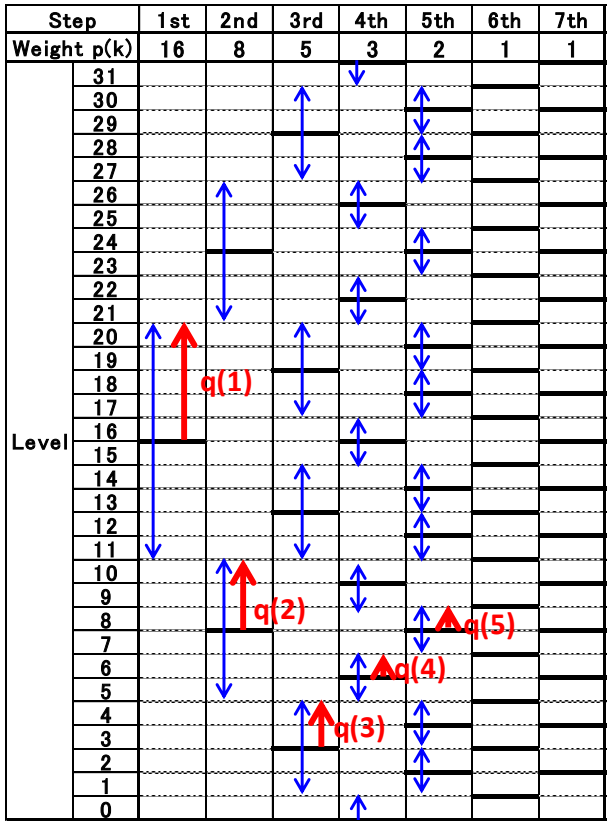
従来手法

1.70進数



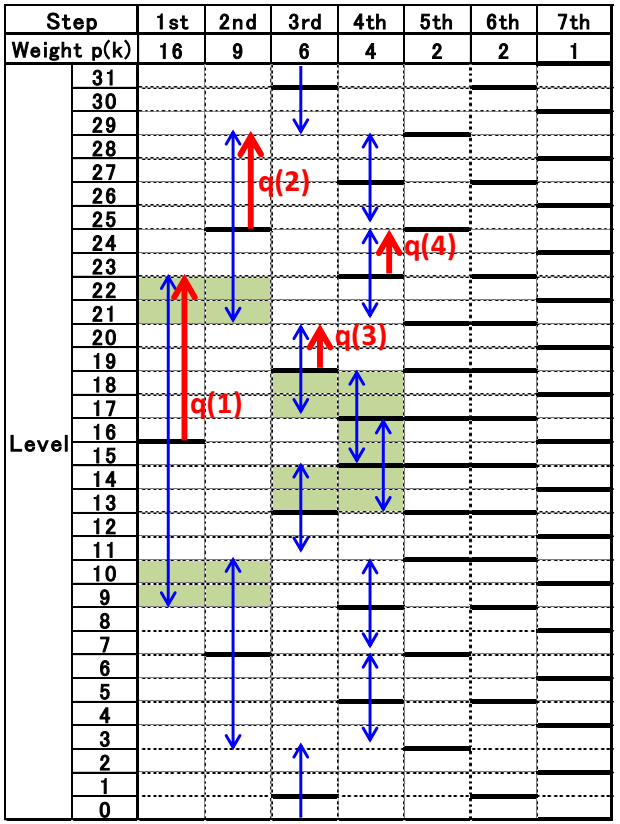
提案手法

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

Outline

- 研究目的と概要
- SAR ADCの冗長設計
- 補正能力の一般化と従来設計手法
- フィボナッチ数列を用いた冗長設計SAR ADC
- 整数論を用いた冗長設計SAR ADC
  - Nーボナッチ数列
  - リュカ数列
- まとめ

n-ボナッチ数列

トリボナッチ数列(前3項の和)

$$T_0 = T_1 = 0\ ,\ T_2 = 1$$
$$T_{n+3} = T_n + T_{n+1} + T_{n+2}$$

$$\lim_{n\to\infty} \frac{T_n}{T_{n-1}} = 1.839286$$

テトラナッチ数列(前4項の和)

$$T_0 = T_1 = T_2 = 0\ ,\ T_3 = 1$$
$$T_{n+4} = T_n + T_{n+1} + T_{n+2} + T_{n+3}$$

$$\lim_{n\to\infty} \frac{T_n}{T_{n-1}} = 1.927562$$

ペンタナッチ数列(前5項の和)

$$H_0 = H_1 = H_2 = H_3 = 0\ ,\ H_4 = 1$$
$$H_{n+5} = H_n + H_{n+1} + H_{n+2} + H_{n+3} + H_{n+4}$$

$$\lim_{n\to\infty} \frac{H_n}{H_{n-1}} = 1.965948$$

⋮

項が前n個の項の和の数列をn-ボナッチ数列と呼ぶ
これら隣り合う項の比率αは黄金比φ以上で2より小さい

$$1.61803 \leq \alpha < 2$$

⇒ 整数で二進数より遅く、冗長性付加に適する

各数列の共通性質

q(k)について発見されたこと

- 数列の最後の誤差補正値qは必ず1が連続し漸化式の項数

- どの数列も最終から2ステップだけは誤差補正ができない(q=0)

- 第n項までの総和に1を足したものが、
  {n+漸化式の右辺項数}の項重みステップの許容範囲に等しい

整数列として利用できるが
SAR ADCへ応用できる性質がない

探索の必要性
- 新たな性質
- 性質の利用方法

Outline

- 研究目的と概要
- SAR ADCの冗長設計
- 補正能力の一般化と従来設計手法
- フィボナッチ数列を用いた冗長設計SAR ADC
- 整数論を用いた冗長設計SAR ADC
  - Nーボナッチ数列
  - リュカ数列
- まとめ

リュカ数列を用いた冗長設計

リュカ数列

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

Édouard Lucas
(仏：1842-1891)

初めの項を計算すると(リュカ数)

（2,）1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521, 843, 1364, 2207, 3571, 5778, 9349, 15127…

※はじめの2を数列の値として入れないこともある

また隣り合う項の比率は以下の値φに収束する(黄金比)

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

フィボナッチ数列以外で黄金比を実現できる唯一の数列

リュカ数を用いた冗長設計

冗長設計の境界条件　＝　基数が黄金比φ

黄金比を作ることのできる整数列
⇒フィボナッチ数列とリュカ数列

リュカ数を用いて冗長設計を行うとどうなるか？

リュカ数(2項の和)		
項の値	比率	和
2		2
1		3
3	3	6
4	1.333333	10
7	1.75	17
11	1.571429	28
18	1.636364	46
29	1.611111	75
47	1.62069	122
76	1.617021	198
123	1.618421	321
199	1.617886	520
322	1.61809	842
521	1.618012	1363

作成条件
リュカ数は最初から3項目で項比率が2を一度超える
⇒項の順序を大きさ順に変更する
(補正力は和によって決まるので初期条件の順番は無関係)

リュカ数を用いた冗長設計

リュカ数列SAR ADC

二点の性質を発見！

①補正可能範囲qはリュカ数

②補正可能範囲は必ず接する

⇩ なぜ成り立つか?

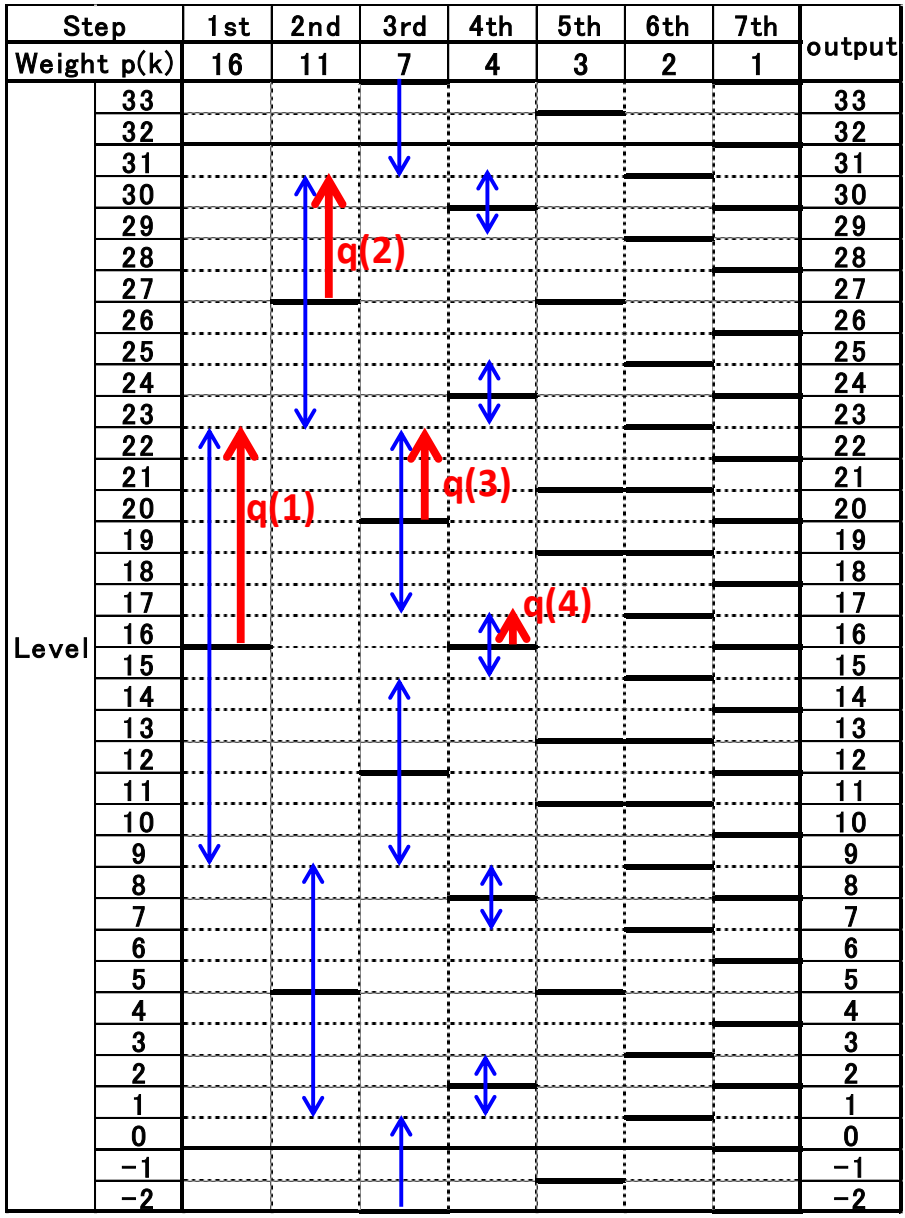
「リュカ数の和」は
フィボナッチ数の和と同じ性質

⇒証明可能

再確認

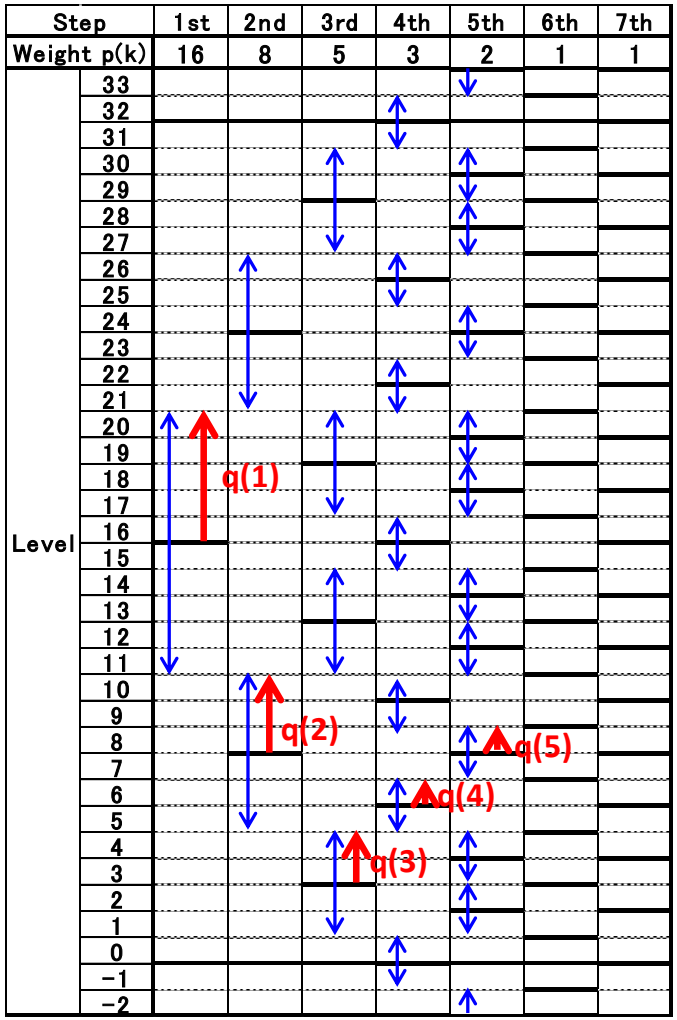
補正可能範囲qが接する

⇒ 基数が黄金比φである

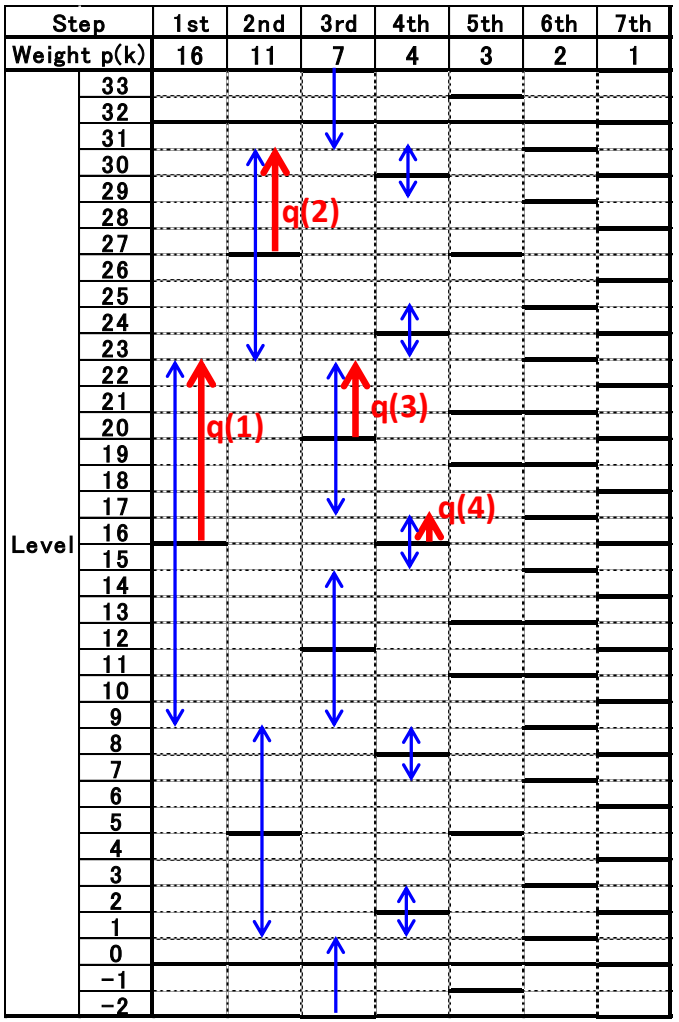


フィボナッチ数列 vs リュカ数列

Fibonacci SAR ADC



Lucas SAR ADC



両者は整数論の応用有効性を示す

まとめ

◆SAR ADC冗長設計の効果と従来手法の問題点を提示
- 適切な分銅p(k)によって能力が決定

◆フィボナッチ数列を用いた冗長設計SAR ADC効果の検証
- もっとも効率の良い設計
- 冗長設計の基準

◆補正可能範囲が接する条件は黄金比であり、
リュカ数でも実現可能

従来手法を改善・従来手法へ貢献

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)
⇒ディジタル信号との相性良し

AD/DA変換器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

質疑応答

Q & A

- 東芝　松野隼也さん（座長）

  確認になるがN-ボナッチ数列は、冗長度の調節に使えるということでいいのか？

  ⇒冗長度の調節に使うことができます。整数列でフィボナッチ数列より冗長度の低い冗長設計が可能です。ただしフィボナッチ数列やリュカ数列のような、特別美しい関係性は発見されていないので、現段階では“整数で一定比率を実現”という性質のみ利用することとなります。

- 東京都市大学　堀田先生

  (感想)面白い話をありがとう。実は我々が研究しているβ変換(パイプライン傾き制御)ADCも非二進数を使う必要がある。そのとき黄金比を使うと打切り誤差が出ないという特性があり、これは整数列で実現ができるからだと考えられる。今後とも非二進数の美しさを突き詰めてほしい。

- 横川電機　加藤さん

  (感想)面白く聞かせてもらった。冗長性というのは機能安全という考え方から産業界で重要な部分となる。こういった原理的な部分から突き詰めてもらえるのは非常に価値があるので期待。

Appendix

付録

実際の使用に向けた検証

一般化非2進SAR ADCアルゴリズムについて

実際の使用に向けて以下２点を検証

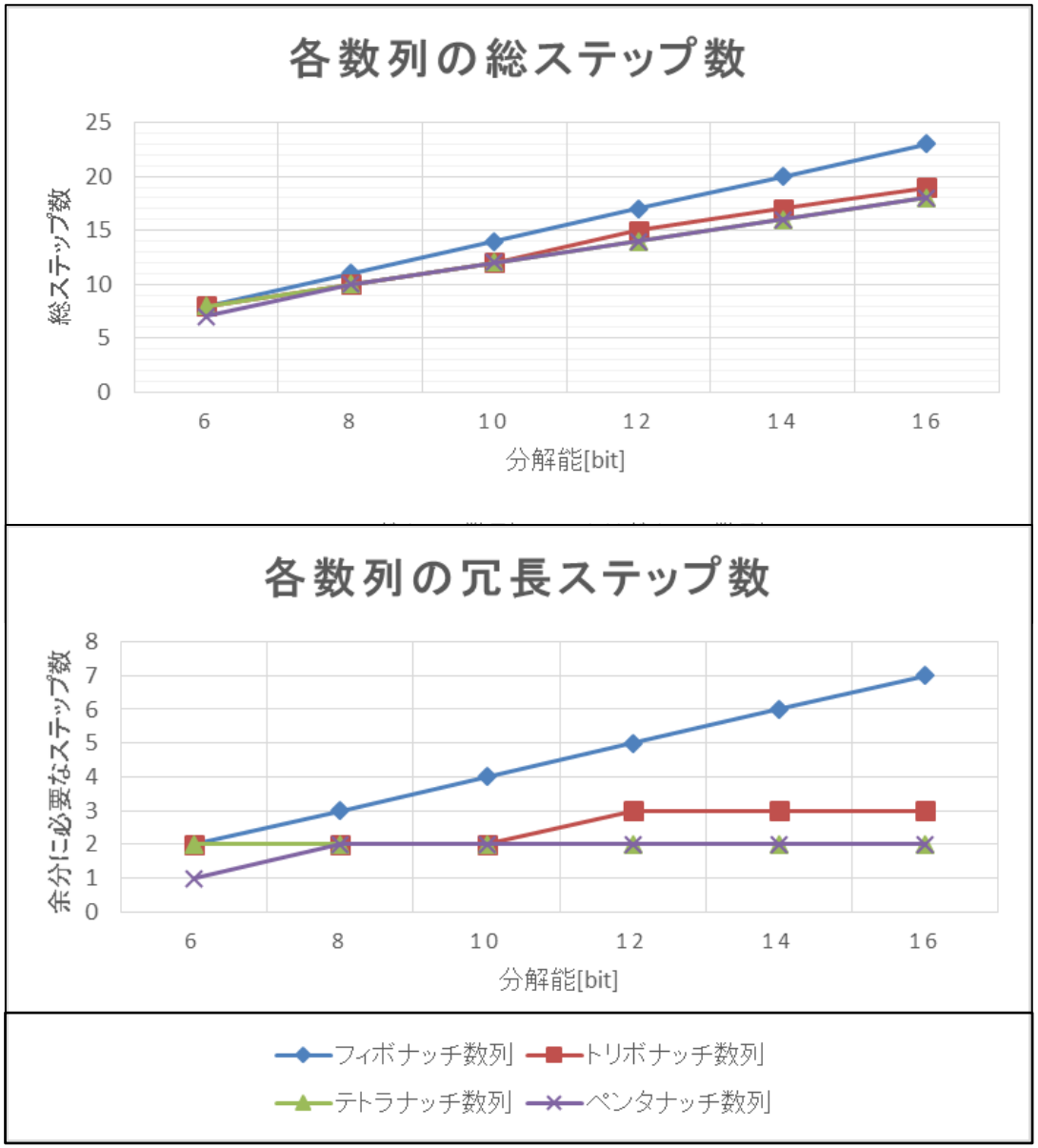
- 判定ステップ数(冗長ステップ数)
- 各ステップの誤差補正許容値

(分解能:6bit,8bit,10bit,12bit,14bit,16bit)

SAR ADC冗長設計に使用できるn-ボナッチ数列を調査

- フィボナッチ数列(前2項の和) ⇒ 約1.62進
- トリボナッチ数列(前3項の和) ⇒ 約1.84進
- テトラナッチ数列(前4項の和) ⇒ 約1.93進
- ペンタナッチ数列(前5項の和) ⇒ 約1.97進

総判定ステップ数・冗長ステップ数

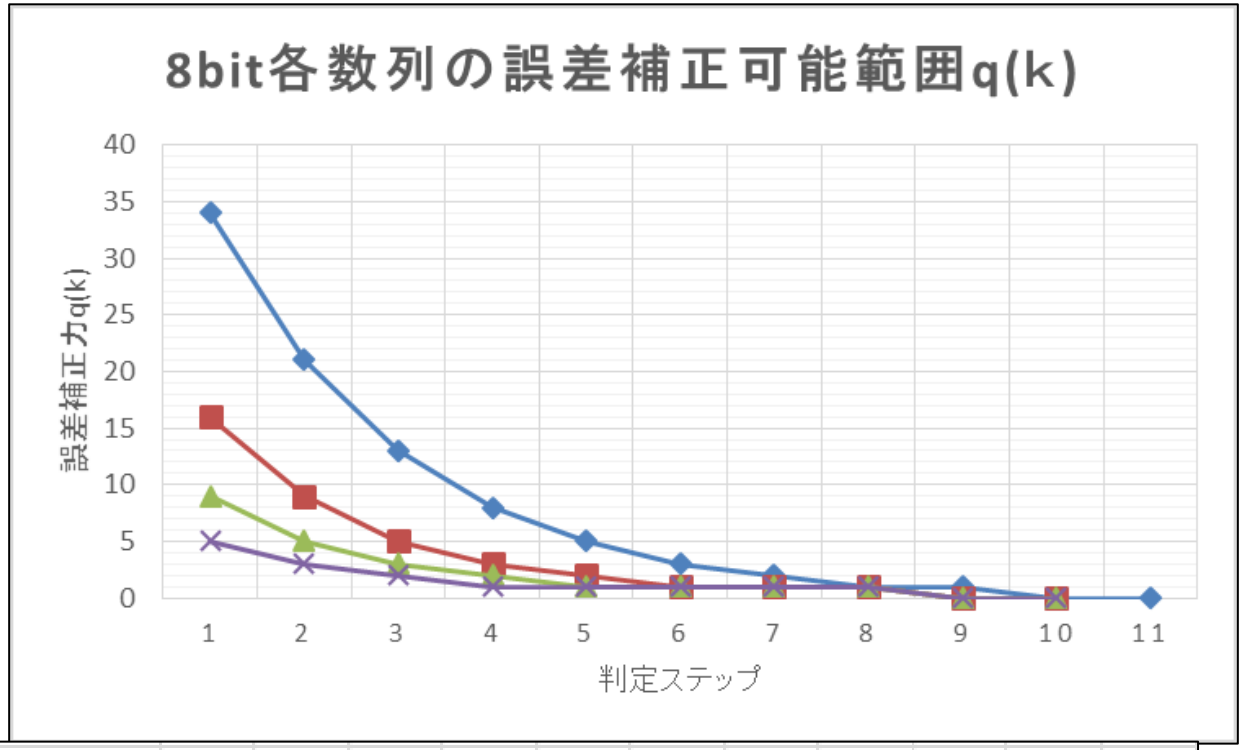


Nbit分解能における判定ステップ数n

$$\frac{2^N}{2} \leq 1 + \sum_{i=1}^{n} X_i$$

各数列と総ステップ数
の相関性無し
(二進数との関係性が希薄)

誤差補正可能範囲q（数列による違い）



q(k)値

フィボナッチ数列	34	21	13	8	5	3	2	1	1	0	0
トリボナッチ数列	16	9	5	3	2	1	1	1	0	0	
テトラナッチ数列	9	5	3	2	1	1	1	1	0	0	
ペンタナッチ数列	5	3	2	1	1	1	1	1	0	0	

※フィボナッチ数列のみ11step、他は10step

誤差補正可能範囲q発見事項

qの数値から発見されたこと

- 数列の最後の誤差補正値qは必ず1が連続し回数はフィボナッチ数列は2回、トリボナッチ数列は3回など漸化式の項数である
- どの数列も最終から2ステップだけは誤差補正ができない(q=0)
  ⇒重みは必ず最後1,1となることから明らか
- 第n項までの総和に1を足したものが、{n+漸化式の右辺項数}の項重みステップの許容範囲に等しい
  ⇒qの式と漸化式の関係から明らか

q(k)値

	1	2	3	4	5	6	7	8	9	10	11
フィボナッチ数列	34	21	13	8	5	3	2	1	1	0	0
トリボナッチ数列	16	9	5	3	2	1	1	1	0	0	
テトラナッチ数列	9	5	3	2	1	1	1	1	0	0	
ペンタナッチ数列	5	3	2	1	1	1	1	1	0	0	

誤差補正可能範囲q発見事項

qの数値から発見されたこと

- 数列の最後の誤差補正値qは必ず1が連続し回数はフィボナッチ数列は2回、トリボナッチ数列は3回など漸化式の項数である
- どの数列も最終から2ステップだけは誤差補正ができない(q=0)
  ⇒重みは必ず最後1,1となることから明らか
- 第n項までの総和に1を足したものが、
  {n+漸化式の右辺項数}の項重みステップの許容範囲に等しい
  ⇒qの式と漸化式の関係から明らか

q(k)値

	1	2	3	4	5	6	7	8	9	10	11
フィボナッチ数列	34	21	13	8	5	3	2	1	1	0	0
トリボナッチ数列	16	9	5	3	2	1	1	1	0	0	
テトラナッチ数列	9	5	3	2	1	1	1	1	0	0	
ペンタナッチ数列	5	3	2	1	1	1	1	1	0	0	

誤差補正可能範囲q発見事項

qの数値から発見されたこと

- 数列の最後の誤差補正値qは必ず1が連続し回数はフィボナッチ数列は2回、トリボナッチ数列は3回など漸化式の項数である
- どの数列も最終から2ステップだけは誤差補正ができない(q=0)
  ⇒重みは必ず最後1,1となることから明らか
- 第n項までの総和に1を足したものが、{n+漸化式の右辺項数}の項重みステップの許容範囲に等しい
  ⇒qの式と漸化式の関係から明らか

q(k)値	1	2	3	4	5	6	7	8	9	10	11
フィボナッチ数列	34	21	13	8	5	3	2	1	1	0	0
トリボナッチ数列	16	9	5	3	2	1	1	1	0	0	
テトラナッチ数列	9	5	3	2	1	1	1	1	0	0	
ペンタナッチ数列	5	3	2	1	1	1	1	1	0	0	

誤差補正可能範囲q発見事項

qの数値から発見されたこと

- 数列の最後の誤差補正値qは必ず1が連続し回数はフィボナッチ数列は2回、トリボナッチ数列は3回など漸化式の項数である
- どの数列も最終から2ステップだけは誤差補正ができない(q=0)
  ⇒重みは必ず最後1,1となることから明らか
- 第n項までの総和に1を足したものが、{n+漸化式の右辺項数}の項重みステップの許容範囲に等しい
  ⇒qの式と漸化式の関係から明らか

q(k)値	1	2	3	4	5	6	7	8	9	10	11
フィボナッチ数列	34	21	13	8	5	3	2	1	1	0	0
トリボナッチ数列	16	9	5	3	2	1	1	1	0	0	
テトラナッチ数列	9	5	3	2	1	1	1	1	0	0	
ペンタナッチ数列	5	3	2	1	1	1	1	1	0	0	

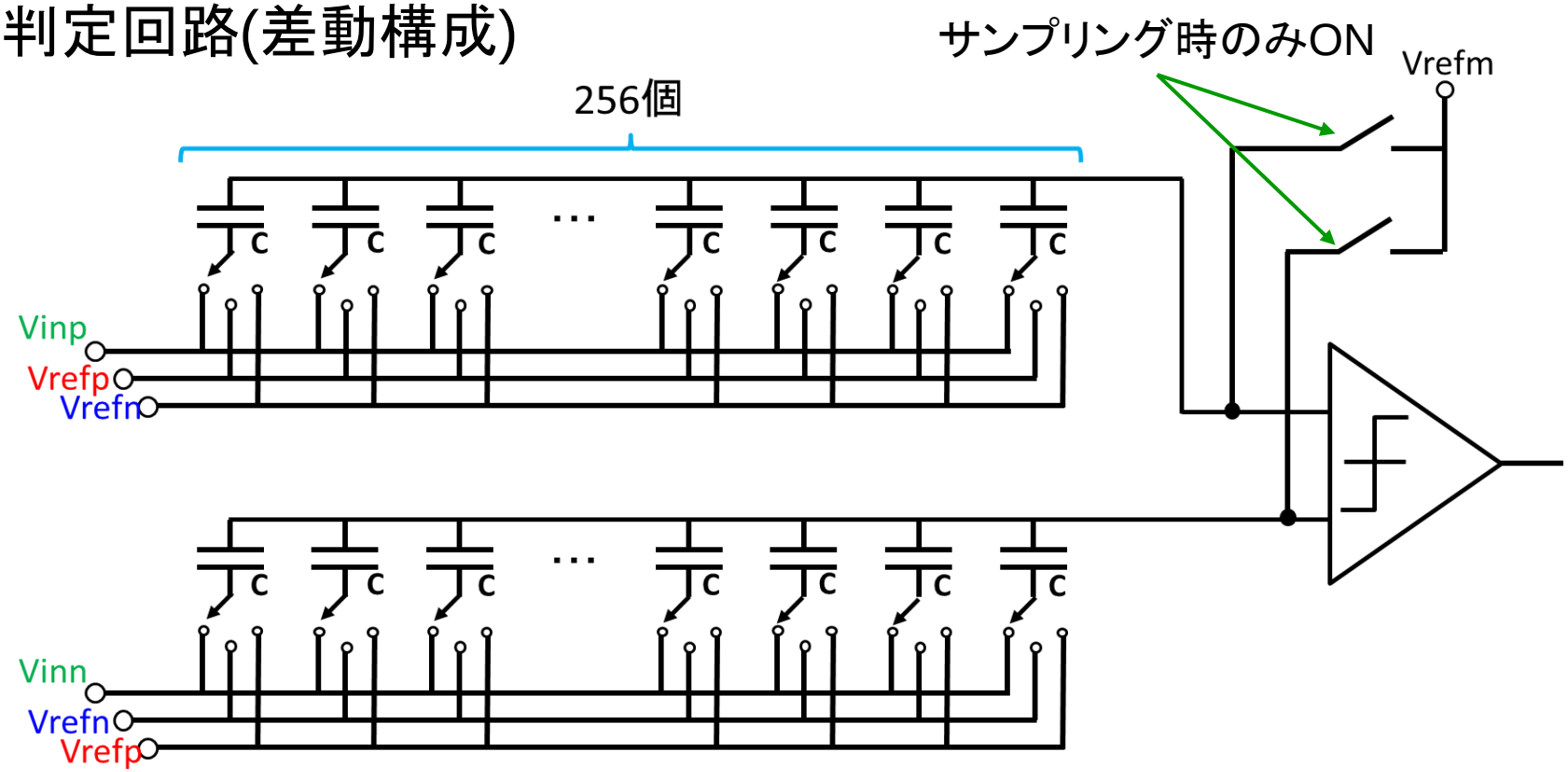
+1

+1

トリボナッチ(3項の和)		
項の値	比率	和
0		0
0		0
1		1
1	1	2
2	2	4
4	2	8
7	1.75	15
13	1.857143	28
24	1.846154	52

比較判定部（C-array and Comparator）

比較判定回路(差動構成)



上段の動作

サンプリング：Vinp ⇒ $\frac{n}{2^N}[LSB]$との比較：n個Vrefp , $(2^N - n)$個Vrefn

下段の動作

サンプリング：Vinn ⇒ $\frac{n}{2^N}[LSB]$との比較：n個Vrefn , $(2^N - n)$個Vrefp