

# フィボナッチ数列を使う電子回路の話し

小林 春夫、小林 佑太郎 群馬大学大学院 理工学府 電子情報部門

## 1. フィボナッチ数列

フィボナッチ数列とは

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

ただし  $F_0 = 0, F_1 = 1$

の漸化式で定義される数列である。1202年にイタリアの数学者レオナルド・フィボナッチが発行した『算盤の書』(Liber Abaci)に記載された数であり、整数論の代表とも言える。いくつかの項を計算すると以下の値となる。

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, …

すなわち隣り合う二項の和が次の項になる数列で、この整数値はフィボナッチ数と呼ばれる。

また隣り合う二つのフィボナッチ数の比率はある値へ収束する。

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

この収束比率 $\varphi$ は“黄金比”と呼ばれ、古代より視覚的に落ち着く最も美しい比率であるとされている。<sup>[1-4]</sup>

これらのフィボナッチ数や黄金比は上記のように簡単かつ単純に導くことができるが、動物・植物・美術など様々な場所に現れること、多くの独特な性質・特徴を持っていることなどから不思議な数として認識されており、フィボナッチの発表から800年以上が経った現在でも研究が続けられている。算額問題でも整数解がこの数に関係することがある。

## 2. AD変換と二進数

最近では、自動車のエレクトロニクス化が進み、車載システムのマイコンと組み合わせて使用するアナログ→デジタル(Analog - to - Digital: AD)変換器の高性能化が進んでいる。

通常、デジタル化とはある量を2進数、10進数等での整数値で表わすことである。例えば43.12ボルトの入力電圧があったとして、43という整数値が簡単に得られる訳ではない。これには32.0, 16.0, 8.0, 4.0, 2.0, 1.0Vの基準電圧をあらかじめ用意して置き、まず32Vよりも大きい小さいかで比較する。大きければ1、小さければ0とする。次に43.12-32.0=11.12Vを16.0Vと比較する。この場合は小さいので0とするが、8.0Vよりは大きいので1とする。11.12-8.0=3.12は4.0Vより小さいので0とするが、2.0Vよりは大きいので1、最後に3.12-2.0=1.12、つまり

$$1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 = 43$$

$$1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2 + 1 = 43$$

この6bitの係数(1,0,1,0,1,1)のデジタル(整数値)出力を得るのがいわゆるAD変換である。10進数で表せば43という整数値が得られる。

## 3. 逐次比較近似AD変換

ところで、広く用いられているAD変換器の構成として逐次比較近似があり、その基本動作は“天秤の原理”を用いたもので、通常、二進数のアルゴリズムが用いられる。

図1に4bitの2進数を求める解探索動作模式図を示す。

前述の基準電圧(重みweightという)は8,4,2,1Vの場合に相当する。図1でstep1の横線は8のweight(錘、基準電圧)が有る無しの境界を意味し、ある場合は上側、ない場合は下側となる。次にstep2の横線はstep1の上下それぞれに4のweightで有る無しを、step3の横線

Step	1	2	3	4	output
Weight	8	4	2	1	
LEVEL 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

図1 4-bit SARADC の二進探索アルゴリズム

step	1	2	3	4	誤判定なし
weight	8	4	2	1	output Code
L 15					15 1111
14					14 1110
13					13 1101
12					12 1100
11					11 1011
10					10 1010
9					9 1001
8					8 1000
7					7 0111
6					6 0110
5					5 0101
4					4 0100
3					3 0011
2					2 0010
1					1 0001
0					0 0000

図2 二進探索アルゴリズムと出力コード

は step2 の区切りに 2 の weight で有るなし、最後の step4 は 1 の weight で有る無しを意味している。

今、7.3 の入力 (LEVEL) はあったとする。step 1 で 8 の錘と比べると小さいので下側へ、step2 の 4 の錘と比べる。大きいから上側へ、step3 の 4~8 間の 2 の錘と比べると上へ、最後に step4 で 1 の錘で比べると上へ、結果的に 7 の出力となる。矢印は、基準錘との比較の様子を示すので、「天秤の原理」と呼ばれる。矢印の下降は基準錘より軽い、つまり無いことを意味し、上昇は錘があることを示す。つまり、

$$0 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 = 7$$

$$0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2 + 1 = 7$$

により 4bit の 2 進係数 (0,1,1,1) が得られる。以上を逐次比較近似 AD 変換という (図2)。

図1の変換アルゴリズムに基づく AD 変換器の主な電子構成要素は図3のようであり、入力電圧がサンプル&ホールド回路で得られ、比較がコンパレータ、基準電圧が逐次比較レジスタ (SAR) と DA 変換で得られる。それとサンプリングのクロックの五つである。正確な変換をするためには、比較結果に直接影響するサンプル&ホールド回路と DA 変換器に精度が必要である。

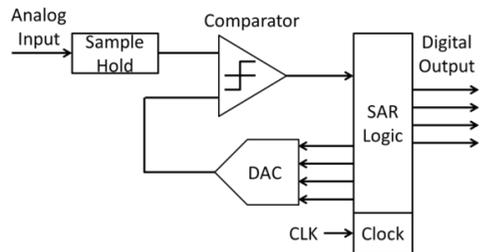


図3 逐次比較近似 AD 変換器の構成

この逐次比較近似 AD 変換器は高分解能(8~18bit)、中速サンプリング(5MSps 程度)の AD 変換器である。低消費電力かつ小チップ面積であるという特徴から、自動車や工業用機械などで広く用いられている。

一般的に比較電圧の大きさは、二進数の重みの加減算から構成される数字を使って比較しており、時間(ステップ数)的に無駄がなく効率が一番良いとされる。しかしながら実際の動作においては入力信号の急激な変動(ノイズなど)や基準電圧などの原因により、比較の際に出力を誤ってしまう可能性がある。二進探索アルゴリズムの場合、一度でも判定を誤るとその二進数の性質(十進数と二進数が一対一対応する)から正しい出力を得ることができなくなる。例えば図1で 7.3V を 8.以上と誤判断すると、図1で step 1 は 8、step2 は 12、step3 は 10、step4

は9となり、7ではなく、9と出力してしまう。そこで、このような判定の誤りを自ら修正する方法として時間冗長性を用いた逐次比較近似AD変換器というのがある。<sup>[5]</sup> 時間冗長性というのは、図3の比較ステップ回数を増やす、という意味で用いている。

従来の冗長設計方法では、設計者が任意に決めた基準値の使用によって補正力にばらつきが生まれ、性能向上を妨げていた。そこで有名な整数論であるフィボナッチ数列を基礎とする設計を考えた訳である。特にフィボナッチ数列が約1.62進数を整数のみで実現できるという“黄金比”という性質を持っていることに着目した。

#### 4. 冗長型逐次比較近似AD変換器 (非2進数型)

この方式は二進数でない重み比較電圧を用いる点にある。この動作例を図4に示す。比較電圧重み(基準電圧)を大きいほうから順に8, 6, 3, 2, 1とした場合に、ある入力電圧が正しく判定されるか否かを示している。図4における二例の違いは、入力8.3に対して1step目で8以上と正判断する場合と、8より小さいと誤判断する場合を示す。判定で正誤があっても、後段のステップで補正されて二例とも正しい変換結果8を得られる。つまり、

正判定は 8 → 14 → 11 → 9 → 8  
 $8 + 6 - 3 - 2 - 1 = 8$   
 デジタルコード=8

誤判定は 0 → 2 → 5 → 7 → 8  
 $8 - 6 + 3 + 2 + 1 = 8$   
 デジタルコード=8

比較回数が一回増えると出力デジタルコードが1bit増えて、表現可能な値の種類は2倍になり、ひとつの出力値を複数のデジタルコードで表現するために誤りを補正することができる。これがデジタル誤差補正の考え方であり、AD変換の信頼性向上に貢献する。さらに誤判定を後段で補正することができるので、

Step	1	2	3	4	5	output
Weight	8	6	3	2	1	
LEVEL						
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
-1						-1

図4 4-bit 5-step SAR ADCの非二進探索アルゴリズム

step	1	2	3	4	5	output	誤判定なし
weight	8	6	3	2	1		Code
16						16	11011
15						15	11010
14						14	11001
13						13	10110
L 12						12	10101
E 11						11	10100
V 10						10	10011
E 9						9	10010
L 8						8	10001
7						7	01110
6						6	01101
5						5	01100
4						4	01011
3						3	01010
2						2	01001
1						1	00110
0						0	00101
-1						-1	00110

図5 非二進探索アルゴリズムと出力コード

DA変換器出力が完全に安定化(整定)するまでの時間が不要となり、判定回数を増やしたとしても全体の変換速度を向上することができる。このように冗長設計によって信頼性と変換速度に優れたAD変換器を作ることが可能である。逐次比較近似AD変換器の冗長設計を一般化すると、k step目で誤判定を起こしたとしても補正をすることが可能な入力範囲差 $q(k)$ を決定することができる。

Step	1	2	3	4	5	output
Weight	8	6	3	2	1	
LEVEL	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
-1					-1	

図 6 4-bit 5-step の冗長探索アルゴリズムと誤差補正範囲  $q(k)$  の図表示

図 4 の例での誤差補正可能範囲を図示したものを図 6 に示す。図 6 中で両矢印が誤差補正可能な入力の範囲であり、上矢印が  $q(k)$  を表現している。図 6 からわかるように  $q(k) \geq |V_{ref}(k) - V_{in}|$  を満たすと  $k$  step 目で判定誤りを生じたとしても、後の判定を誤らなければ正しい値へと修正することが可能である。すなわち各ステップの  $q(k)$  の大きさが逐次比較近似 AD 変換器の能力の高さを示すことになる。また  $q(k)$  は  $k$  step 目の比較電圧の重み weight  $w(k)$  によってのみ決定されるため、各ステップの比較電圧重みが冗長設計における最も重要なパラメータとなる。

#### 従来手法の比較電圧重み決定方法

逐次比較近似 AD 変換器の冗長設計効果の大きさは比較電圧重み  $w(k)$  によってのみ決定されるので、たとえ回路規模等を増大させて冗長設計を施したとしても、比較電圧重み  $w(k)$  の選択が適切にされないと補正効果を得ることができない。比較電圧重みは隣り合う重みの比率が 1 以上かつ 2 以下である必要がある。すなわち二進数列から単項列の間の比率で進む重みの利用

が必要である。開発者が数列のそれぞれの項を任意に決定することもできるが、一般的な従来手法は  $k$  step 目の比較電圧重み  $w(k)$  を適切な範囲内の基数  $x$  (radix) を用いた基本式から決定する。

#### 従来手法を用いた冗長設計の問題点

従来手法で冗長設計することは可能であるが、ここにはいくつかの問題点がある。問題点のひとつには  $1 < x < 2$  の基数に基づく冗長アルゴリズムでは比較電圧重み  $w(k)$  が整数にならないことがある。AD 変換器に使用する比較電圧重みは変換精度や設計容易化のために整数である必要があるため、比較電圧重み  $w(k)$  の決定には整数値への丸めを行う必要がある。このような端数処理は隣り合う項の比率(進数)の変化を生じ各ステップにおける補正力  $q(k)$  にばらつきを生じ、性能向上を妨げる。

また更なる問題点として基数決定の難しさがある。図 4 で示した例は基数  $x$  が 1.80 として各値に四捨五入を使用した場合であるが、図 5 を見てみると補正可能な範囲を示す両矢印がすべての入力範囲を埋め尽くしていないということがわかる。このような基数の決定をすると AD 変換器の入力によっては、原理的に補正が不可能になる。図 5 の例で言うと AD 変換器の入力が 1~3, 7~9, 13~15 [LSB] の範囲外の場合は補正することが不可能であり、不適切な基数の決定が補正力の弱体化につながるということがわかる。この問題に対して  $q(k)$  の増加を狙い冗長度を大きくして基数  $x$  を小さく設定すると、総ステップ数  $M$  が増加して変換速度が低下する。逐次比較近似 AD 変換器の冗長設計は補正力の大きさと変換速度がトレードオフの関係にあり、設計者は最も適する基数  $x$  の探索を強いられる。これらの問題は従来手法において必ず発生するものであり、冗長設計における性能低下や設計難度上昇を引き起こす原因である。

#### 5. 整数論を用いた冗長アルゴリズムの設計

冗長設計には優れた AD 変換器設計の可能性

があるが、従来の手法では十分な効果の発揮が難しいことを示した。そこで従来手法に取って代わる冗長設計方法を検討し、整数を扱う AD 変換器に適した整数論を応用することを提案する。中でも今回は特に良く知られている整数論のひとつである“フィボナッチ数列”を用いた冗長設計方法を示す。

### 逐次比較近似 AD 変換器のフィボナッチ数列の応用

フィボナッチ数列は整数のみで構成される数列であり、隣り合う項の比率は黄金比の約 1.62 になる。このことを別の表現にすると、端数処理がまったく使われていない整数のみで、基数が黄金比である約 1.62 進の数字列を実現できるということになる。一般的に整数の小数倍は小数となるが、フィボナッチ数列は整数の小数倍(約 1.62 倍)が整数となるために冗長設計に応用できるのである。整数項で一定比率を保持することができるフィボナッチ数列の強力な性質を用いて逐次比較近似 AD 変換器の設計を行う。

⑥ 逐次比較近似 AD 変換器へ冗長性を与えるために比較電圧重み  $w(k)$  の変更を行う。

フィボナッチ数列を組み合わせた冗長設計利用 4bit6step の逐次比較近似 AD 変換器の解探索動作を図 7, 8 に示す。最初のステップはハーフスケールを選択し、以降はフィボナッチ数重みで実現している。図 7 において両矢印が後段での補正が可能な入力誤差範囲を示し、上矢印が誤差許容値  $q(k)$  である。

### フィボナッチ数列の冗長応用の性質

図 5 から理論的に興味深い性質を 2 つ発見できる。以下にそれら性質を示す。

#### 性質①

誤差許容値  $q(k)$  は必ずフィボナッチ数となり、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。

#### 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。すなわ

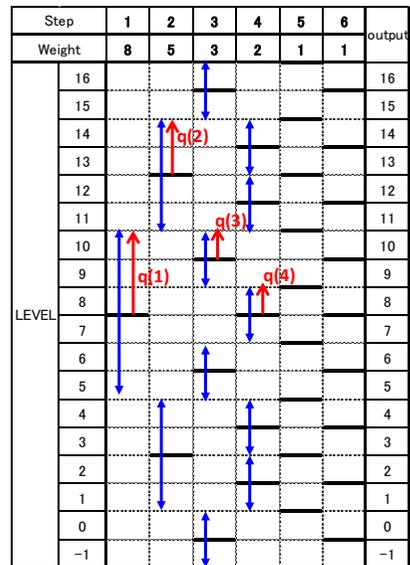


図 7 4-bit 6-step のフィボナッチ数列を用いた逐次比較近似 AD 変換器の冗長探索アルゴリズム

step	1	2	3	4	5	6	output	誤判定なし
weight	8	5	3	2	1	1		Code
16							16	111001
15							15	110110
14							14	110101
13							13	110010
L 12							12	101101
E 11							11	101010
V 10							10	101001
E 9							9	100110
L 8							8	100101
7							7	011010
6							6	011001
5							5	010110
4							4	010101
3							3	010010
2							2	001101
1							1	001010
0							0	001001

図 8 フィボナッチ数探索アルゴリズムと出力コード

ち図 5 において  $k$  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、もしくは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。

## フィボナッチ数列の冗長応用の有効性

フィボナッチ数列を逐次比較近似 AD 変換器へ応用することで二つの性質が得られたが、そのうち性質②の発見の意義と重要性はきわめて大きく、その理由が二つある。

まず一つ目にこの性質が逐次比較近似 AD 変換器における冗長設計のあらゆる基準となることが挙げられる。フィボナッチ数列を利用した冗長設計では基数  $x$  が黄金比(約 1.62 進)であることに等しく、補正可能範囲が接する条件となっていた。このことは黄金比の基数を基準とすれば誤判定補正可能な範囲  $q(k)$  が重なるか離れるかが判定できるということを示している。すなわち基数  $x$  の値が黄金比(約 1.62)より大きければ、冗長度が小さく図 4 のように  $q(k)$  は離れることになる。また基数  $x$  の値が黄金比より小さければ冗長度が大きく  $q(k)$  は重なることになる。このようにして黄金比を冗長度の基準とすれば、補正力の大きさから基数  $x$  を容易に決定できる。

また二つ目の理由としてフィボナッチ数列を用いた冗長設計は最も効率のよい設計ができることが挙げられる。性質②が成り立つ限り補正可能範囲  $q(k)$  は重なることなく接するので、最小の冗長度で補正できない入力範囲が存在しないということになる。すなわち整数で構成されるフィボナッチ数を比較電圧重み付けに利用することで、最小ステップ数で全入力範囲を補正可能な無駄がない設計を実現できるのである。以上の二点からは従来法の問題点を改善できることを示すことができ、フィボナッチ数列を設計に用いることの有効性が確認できた。

AD 変換器の内部回路設計の詳細に立ち入る必要があるため結果のみ記すが フィボナッチ重み付け探索アルゴリズムを用いると AD 変換器が高速になり、内部の DA 変換器 (DAC) の構成もシンプルになることが示せている。<sup>[6]</sup>

## 6. 更なる整数論の応用

整数論の AD/DA 変換器応用は研究がほとんど行われていない。本論文は整数論の電子回路設計への応用可能性を提示したひとつの例である。これまで整数論のひとつであるフィボナッチ数列を用いた冗長設計方法について述べ、効果を実証することができたが、他の整数論が応用できる可能性も十分あることも付記しておく。<sup>[7]</sup> 筆者の研究室では幾人かの学生とともにこの研究に継続的に取り組んでいく。

### 参考文献

- (1) A. S. Posamentier, I. Lehmann (著), 松浦俊輔 (訳): 「不思議な数列フィボナッチの秘密」, 日経 BP 社 (2010 年 8 月).
- (2) マリオ・リヴィオ (著) 斎藤隆央(訳) 「黄金比はすべてを美しくするか」 早川書房 (2005 年 12 月)
- (3) 渡邊泰治 「黄金比の謎」 化学同人 (2007 年 1 月)
- (4) 桜井進 「雪月花の数学」 祥伝社 (2006 年 7 月)
- (5) 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”, IEICE Trans. Fundamentals, vol.E93-A, no.2, (Feb. 2010).
- (6) Y. Kobayashi, S. Shibuya, T. Arafune, S. Sasaki, H. Kobayashi, “SAR ADC Design Using Golden Ratio Weight Algorithm”, 15th International Symposium on Communications and Information Technologies, Nara, Japan (Oct. 2015).
- (7) 小林佑太郎, 荒船拓也, 渋谷将平, 小林春夫, “疑似白銀比重み付け逐次比較近似 AD 変換器の検討”, 電気学会 電子回路研究会, ECT-16-044, 東京都市大学 (2016 年 3 月).