

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)

ADC/DAC Redundancy Design Using Fibonacci Sequence

Yutaro Kobayashi*, Masaki Kazumi

Yang Zhixiang, Haruo Kobayashi (Gunma University)

This paper describes redundancy design of ADCs/DACs and their calibration method using Fibonacci sequence to realize accurate and/or high-speed conversion. We expect that the proposed method can enjoy properties of Fibonacci sequence such as the closest terms ratio called “golden ratio” and realization of all terms with integers. We introduce theoretical basis to improve performance of SAR ADCs with time redundancy and current-steering DACs with space redundancy utilizing Fibonacci sequence properties.

キーワード：フィボナッチ数列, ADC, DAC, 冗長性, 校正

(Keywords: Fibonacci sequence, ADC, DAC, redundancy, calibration)

1. はじめに

エレクトロニクス分野ではデジタル信号処理技術が著しく進歩しつつある。実世界/自然界のアナログ信号と LSI 上でのデジタル信号の相互の変換の役割を果たす ADC (Analog to Digital Converter) と DAC (Digital to Analog Converter) がますます重要になってきており、現代の情報処理や計測制御において欠かすことのできない重要な役割を担っている。様々な精密機器が存在する現代において、これらの高精度化・高速化の要求が著しい。

ADC の方式にはさまざまな種類が存在するが、ここでは産業界で広く使用されているコンパレータを一つ用いる逐次比較型(SAR)の ADC について記述する。特に時間冗長性を用いてデジタル誤差補正を行う ADC の非二進探索型アルゴリズムをフィボナッチ数列の性質を用いて設計することを検討する。

また適度な冗長性を用いた DAC としてフィボナッチ数で重み付けした電流源 DAC を検討し、自己校正やダイナミックマッチングにより mismatches の影響を周波数拡散する方式を提示する。

2. 逐次比較型 ADC の構成と動作

逐次比較型 ADC (SAR ADC) は天秤の原理を利用してアナログ値をデジタル値へ変換する。図 1 に構成図を示す。SAR ADC は回路の大部分がデジタル回路で構成されるので微細 CMOS での実現に適するため現在注目されている。産業界で広く使用され、車載用マイコン、工業用制御装置などに利用される。他の方式の ADC と比べた際

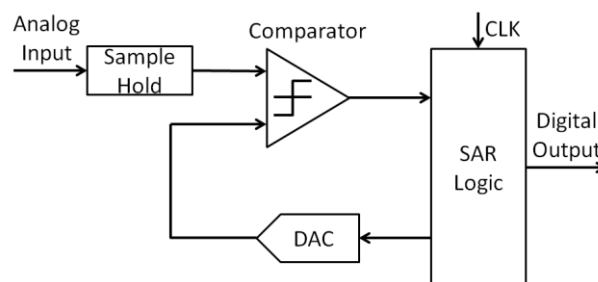


図 1. 逐次比較近似型 ADC の構成図.

Fig.1 Block diagram of an SAR ADC.

の特徴として高分解能・低消費電力・小型、小チップ面積・中速が挙げられる [1]。

動作原理は、アナログ入力をサンプルホールド回路で保持し、コンパレータ (天秤) を用いて基準電圧 (重り) と比較する。比較の結果 (どちらが大きい) から SAR ロジック回路を通すことで 0 か 1 のデジタル出力をする。さらに結果に応じて基準電圧の大きさを SAR ロジック回路で変更、DAC でアナログ信号へ戻しまた比較する…の様にこれらの操作を繰り返してアナログ信号をデジタル信号へ変換する方式である。

図 2 に 4bit (16 レベル) の二進探索アルゴリズム逐次比較型 ADC の出力方法の例を示す (基準電圧の大きさ: 1,2,4,8)。図 2 では太線に従って比較を行っていく。一般的な基準電圧の大きさは二進 (2 の倍数) とされており 1, 2, 4, 8, 16, 32 … という数字を使って比較している。時間的に全く無駄がなく、効率は一番良いとされる。

回数	1	2	3	4
比較電圧	8	4	2	1
レベル	15			
	14			
	13			
	12			
	11			
	10			
	9			
	8			
	7			
	6			
	5			
	4			
	3			
	2			
	1			
	0			

図 2. 4-bit SAR ADC の二進探索アルゴリズム。

Fig.2 Binary search algorithm of a 4bit SAR ADC.

しかしながら現実には入力電圧と参照電圧の比較の際に、出力を誤ってしまう可能性がある。そして二進探索アルゴリズムでは一度判定を誤ると、その二進の性質（十進と二進が 1 対 1 に対応する）から正しい出力を得ることはできない。そこでこのような誤りを自己校正する一つの方法として非二進探索アルゴリズムを利用した SAR ADC を考えていく。

3. 非二進型アルゴリズムを用いた逐次比較 ADC

SAR ADC に非二進型探索アルゴリズムを用いることは時間の冗長性を使うことである [1, 2, 3]。つまり余分に時間を利用して、コンパレータの判定回数（ステップ数）を増やすことで判定誤りを補正するのである。

補正を行う方法は、参照電圧を 1 倍か 0 倍して足していく加算型と、参照電圧を 1 倍か -1 倍して足していく加減算型が考えられる。この二つは 2 進数の重みをもつ参照電圧のときは同じ考え方を適用することができるが、冗長性を持たせると出力デジタル値や補正方法に違いが出る。

3.1 加算型(addition-type)

加算型を、比較の結果で参照電圧を足すか・足さないか決めていくという方法と定義する。天秤での重さの計測に例えると、計測物の逆の天秤皿に、錘を載せるか・載せないかを定める。

N bit 分解能の ADC を M step の比較で実現すると考えると、k step 目の参照電圧 $V_{ref}(k)$ と出力値の十進表現 D_o は以下のようなになる。ただし $p(n)$ は n 回目の参照電圧重み、 $d(n)$ は n 回目のデジタル出力によって変わる値を示し、デジタル出力が 1 ならば $d(n)=1$ 、デジタル出力が 0 ならば $d(n)=0$ である。

$$\begin{cases} k = 1 & V_{ref}(k) = p(k) \\ k \geq 2 & V_{ref}(k) = p(k) + \sum_{i=1}^{k-1} d(i)p(i) \end{cases}$$

$$D_o = \sum_{i=1}^M d(i)p(i)$$

この比較電圧 V_{ref} と入力電圧とをコンパレータを用いて比較していく。入力電圧の方が大きな値のときはデジタル値として 1 を、入力電圧の方が小さな値のときはデジタル値として 0 を出力する。デジタル出力結果は D_o となる。

実際の変換例を図 3 図 4 に示す（4bit 分解能 5step、入力電圧 10.3V、参照電圧の重み 8, 5, 3, 2, 1）。ここでは 4bit 分解能を 5 段階で比較しており、1bit を判定として余分に使っている。図 4 では一回目の比較を誤って判断しているが、図 3 も図 4 も出力は正しく表現できており誤差の補正ができていけると言える。

加算型は出力計算の仕方が 1 のデジタル出力を持つ桁の参照電圧重みを足し合わせればよいので簡単であり、理解も容易である。しかしながら、加算しか行わないので、一度入力値を超えてしまうような判断（デジタル値 1 を出力しすぎることを）をしてしまうと補正することができない。加算しか行わないために加算方向の補正範囲は広い。

回数	1	2	3	4	5	
比較電圧	8	5	3	2	1	
レベル	15					
	14					
	13					
	12					
	11					
	10					
	9					
	8					
	7					
	6					
	5					
	4					
	3					
	2	1	0	0	1	0
	1					
	0					

図 3. 4-bit 5-step 加算型 SAR ADC 動作例。

Fig.3 Operation of a redundant search algorithm of a 4-bit addition-type SAR ADC with 5 steps.



図 4. 4-bit 5-step 加算型 SAR ADC 動作例 (誤判定有) .

Fig.4 Operation of a redundant search algorithm of a 4-bit addition-type SAR ADC with 5 steps in case of error judgment.

3.2 加減算型(Addition-subtraction-type)

加減算型を、比較の結果で参照電圧を加算するか・減算するか決めていく方式と定義する。天秤での重さの計測に例えると、二つある天秤皿のどちらに錘を載せるかを定める方法である。測定物と違う天秤皿に錘を載せれば加算、測定物と同じ天秤皿に錘を乗せれば(負の重さを持つ錘を載せることに相当するため) 減算になる。

N bit 分解能の ADC を M step の比較で実現すると考えると、k step 目の参照電圧 $V_{ref}(k)$ と出力値の十進表現 V_o は以下ようになる。ここでも $p(n)$ は n 回目の参照電圧重み、 $d(n)$ は n 回目のデジタル出力によって変わる値を示し、デジタル出力が 1 ならば $d(n)=1$ 、デジタル出力が 0 ならば $d(n)=-1$ である。

$$V_{ref}(k) = \sum_{i=1}^k d(i-1)p(i)$$

$$D_o = 0.5d(M) - 0.5 + \sum_{i=1}^M d(i-1)p(i)$$

$$d(0) = 1$$

加算型と同様に、入力電圧と参照電圧 V_{ref} を比較してデジタル値を出力する。

実際の変換例を図 5 に示す (4bit 分解能 5step、入力電圧 7.3V、参照電圧の重み 8, 5, 3, 2, 1)。図 5 ではコンパレータが判定誤りを起こしても正しい出力ができていることを示している。さらに加算型では補正することができない参照電圧を大きくしてしまう誤り方も補正できていることがわかる。最終的なデジタル値から、 D_o の式を利用したルックアップテーブルを参照すれば正しい出力

が得られる。

加減算型は、加算方向も減算方向も補正ができるという強みがある。しかしながら、計算が分かりづらく、ルックアップテーブルも必要となる。さらに判定を間違えると、正しい出力から遠ざかる方向へ参照電圧を変更させるので補正範囲が狭くなるという欠点がある。



図 5. 4bit5step 加減算型 SAR ADC 動作例 (誤判定含) .

Fig.5 Operation of a redundant search algorithm of a 4-bit addition-subtraction-type SAR ADC with 5 steps in case of error judgment.

3.3 非二進探索アルゴリズムの考察

加算型でも加減算型でも判定誤りの補正ができることが分かる。これは、比較手順を 1step 増やしたことで十進における数字のデジタルの表現方法が数種類できることによる。例えば加算型の 10 という数字が (10010)、(01110) という二つのデジタル値で表現できるために誤りを補正することができるのである。時間の冗長性(余分な判定ステップ)が数字の表現方法を増加させ、それが誤り補正となっている。これが時間の冗長性を利用した自己校正の方法である。

時間の冗長性を利用することで AD 変換時間が遅くなると考えられるが、実際はこちらの方が全体の AD 変換が早くなる可能性がある。それは図 6 の様に自己校正可能なことによって判定の基準を緩くすることができ(前段での DAC 出力が完全に整定するまで待つ必要がない等)、1 ステップの判定の時間を短くすることが可能だからである。

現実の設計においては各種回路の性能・要求精度・判定時間・冗長ステップ数などのバランスをみる必要があるが、非二進探索アルゴリズムは二進アルゴリズムよりも速度と精度に優れた SAR ADC を実現できる可能性がある。

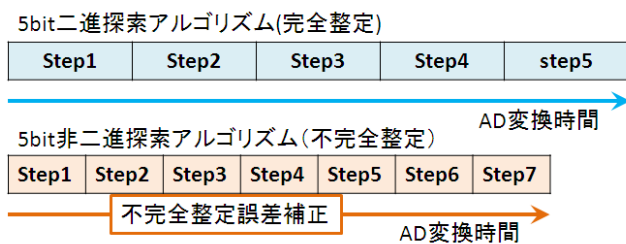


図 6. 5bit SAR AD 変換時間比較.

Fig.6 Comparison of SAR ADC conversion time.

4. フィボナッチを用いた非二進探索アルゴリズム SAR ADC について

3章で示したように非二進型探索アルゴリズムを利用すると高性能 SAR ADC を実現できる可能性がある。冗長性を与えるためには比較に用いられる基準電圧の大きさ(重み)を、二進数より進みの遅い数で決める必要がある。今までは基準の電圧は規則性のある定め方はなくそれぞれの開発者に一任されていた。その定める方法を決めるために、今回はフィボナッチ数列を用いて規則性を持った基準電圧の作り方を考える。

フィボナッチ数列とは以下の漸化式で定義される数列である(1202年にフィボナッチが発行した『算盤の書』(Liber Abaci)に記載された数列[4])。

$$F_0 = 0$$

$$F_1 = 1$$

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

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...

となる。すなわち隣り合う二項の和が次の項になる数列である。

また隣り合う項の比率は以下に収束する(約 1.6 進)。

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

この比率 φ がいわゆる“黄金比(golden ratio)”である。

フィボナッチ数列及び黄金比は上記の様に簡単に導出できるが、非常に多くの特殊な特性(豊富な特徴)を有しており自然界をはじめとして、様々な場面で出現する“不思議な数”である。

フィボナッチ数列は隣り合う数が約 1.6 倍で変わらないために 1.6 進とみなすことができ、今回の非二進探索アルゴリズムの重み付け方法として、このフィボナッチ数列の性質を利用できると我々は考えた。

フィボナッチ数列は適度な冗長性を持っていることに加えて、それを整数で実現できるという強力な性質を保有している。ADC の参照電圧の重さは整数である必要があ

るが、一般的に整数の小数倍は小数となってしまうので、四捨五入を行うと一定比率(一定の進数)を維持することができない。また容量型 DAC を利用する場合は、高精度化のためには容量比は整数であることが望ましい。整数項と一定の項比率を維持することができるフィボナッチ数列は、ADC の参照電圧重み付け方法として非常に有効と考えた。

5. フィボナッチ式重み決定法と考察

この節ではフィボナッチ数列を用いて参照電圧の重み付けの方法を示す(3章の $p(n)$ の導出)。3章で示した通り冗長アルゴリズムで参照電圧の重み付けの理想は、(i) 表現パターンが多い、(ii) ステップ数が少ない、(iii) パターン数に偏在性がない、という三つの基準を満たしていることである。ただし実際の ADC を考えて、以下の条件を満たすものとしている。(a) フィボナッチ数列の特性を用いたものである、(b) レベルの半分は重みとして使用し最初に候補範囲を半分にする、(c) 値が大きくなる場所は必ず漸化式を満たすこと、(d) 重みは整数であること。これらを考慮に入れて考えていく。

現在検討しているフィボナッチ数列を用いた重み付け方法は 3 種類であり、どの方法もフィボナッチ数が保有する性質(フィボナッチ数・黄金比率・漸化式)を利用したものである。以下にその方法を示す。またそれらの方法を利用して 8bit 256 レベル ADC の参照電圧重み項を導出したので共に示す。

① フィボナッチ数列をそのまま用いる方法 (Fibonacci number method)

フィボナッチ数列がすべてのレベルを表現できるところ(項合計が表現レベル数を超えるところ)まで数列を並べていく。これは加算型での利用及び電流源 DAC への応用を考えたもので、必ずしもレベル数の半分が参照電圧となるわけではない。

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 → 8bit 12step
冗長ステップ数 4 項合計 376

② 黄金比を用いる方法 (Golden ratio method)

最上位ビット判別にレベル数の半分を用い、それを黄金比で割った値を小数第一位で四捨五入して項を導出する。四捨五入の結果が 0 となる項が出現するところまで繰り返す。

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

③ 漸化式を用いる方法 (Fibonacci recurrence relation method)

最上位ビット判別にレベル数の半分を用い、その値を黄金比で割り四捨五入で二項目まで導出する。二つの項が作

れるので、漸化式を利用して項を導出していく。この方式を利用すると項の比率が途中で変化を起し、値が発散してしまうので項の比率に範囲を定め、項比率が範囲外となったら黄金比を使って小数第一位を四捨五入することで二項導出し、また漸化式を繰り返す。この方法を項が $1-1=0$ を示すまで繰り返す。実例では項比率 r が $1.5 \leq r \leq 2.0$ のとき補正対象外とした。

1, 1, 2, 4, 6, 10, 16, 32, 48, 80, 128 → 8bit 11step
冗長ステップ数 3 項合計 328

導出の具体的な計算は煩雑となるので省略するが、以上のようにして重みを決定することができる。

DAC への応用や自己補正実現を考慮し、これらに加算型補正方式を適用した場合の各レベル表現パターン数を調べ、図 7・図 8・図 9 のような結果を得た。

結果図から「すべて項合計の半分を中心として左右対称」という特徴があることが分かる。この性質は等比数列に成り立つ性質であるためにそれぞれの方法で 1.6 進の重みを実現することができたと言える。

さらに 3 つの方法がある共通の基準のもと評価したいが、step 数(冗長度)がそれぞれで違うため単純に比較できない。そこで冗長使用率という新たな評価方法を用いる。冗長使用率は、冗長ステップによって増えた表現パターンをどの程度利用できているのかを示すもので、以下に定義式を示す。

$$(\text{冗長使用率}) = (\text{利用範囲内平均表現パターン}) / 2^{M-N}$$

ここで N は bit 数、 M は step 数である。step 数が一段増えるごとに理論上の表現パターンは 2 倍になるので、無駄の少なさを評価できる。

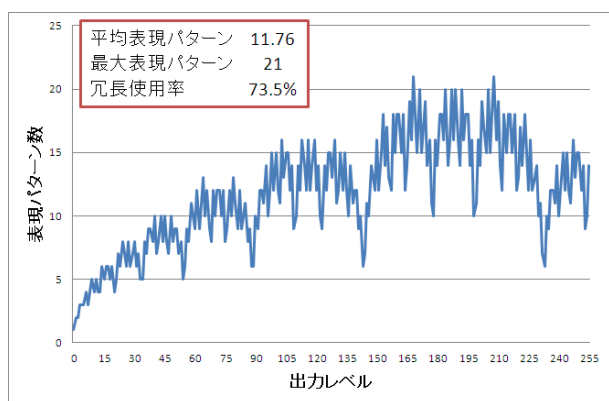


図 7. 加算型表現パターン数(フィボナッチ数法).

Fig.7 Number of combinations in case of Fibonacci number method.

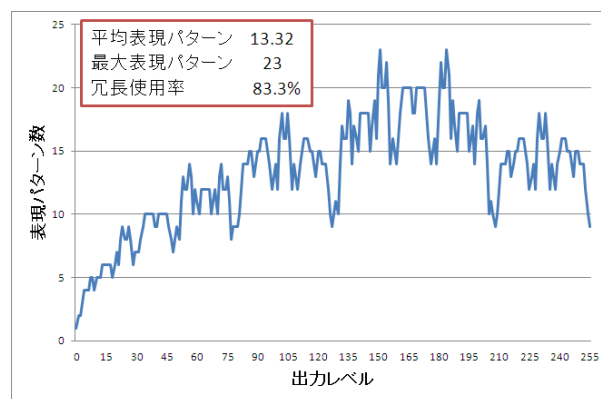


図 8. 加算型表現パターン数(黄金比法).

Fig.8 Number of combinations in case of golden ratio method.

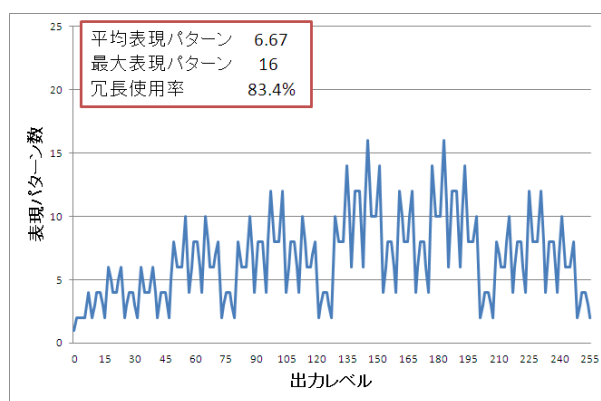


図 9. 加算型表現パターン数(漸化式法).

Fig.9 Number of combinations in case of Fibonacci recurrence relation method.

以上の結果からそれぞれの重み決定法の考察を行う。まず 3 つの方式に共通する特徴を述べる。

加算型において範囲内のパターン数だけで考えるなら全項の合計がレベルに近い方が、範囲外の大きな値(256 ~)の表現が少なく無駄が少なくなるため良い。また項の合計がレベル数に近いほど大きな重みでの判定を間違えられないので高いレベルの表現方法が少なくなると言える。他にも bit 数を増加させるほど bit 数と冗長ステップの比率が小さくなっていくので、これらの補正方法は bit 数の増大によって有利になる。

続いて、各導出方法の特徴を述べる(4bit~8bit)。

① **Fibonacci number method** はフィボナッチ数列をそのまま用いるので導出がわかりやすく、最も忠実に 1.6 進を実現できるため、加算型補正方法や電流源 DAC への応用を考えると一番良いように考えられる。ただしレベルの半分を利用することがほとんどないため、レベルの中心をスタートと考える加減算型の補正方法には向いていない。加えて bit 数と重みに関連性がないため各 bit 数によって冗長ステップ数にばらつきが存在する。

②**Golden ratio method** は冗長 step 数が多くなる傾向がある。導出は単純であるが、四捨五入を利用しているために純粋な 1.6 進にはなりえない。しかしながら、bit 数と項に関係性があるために無駄な表現が少なく、加算型において表現パターンの平均値は①の方式より大きくなることが多い。この方式のみ四捨五入の関係で 1 の項が 3 項出現することもありうるが、その場合は 2 項に変更することを考えてもよい。レベル数の半分を利用できているために加算型・加減算型どちらにも利用できる。

③**Fibonacci recurrence relation method** は冗長ステップ数が少なくなる傾向がある。項導出方法は理解しづらいが、四捨五入はほとんど使われないためにフィボナッチ数列に近い項を出力できる。またレベル数の半分を利用するためにどちらの補正方法も利用することができる。さらに隣り合う項比率の補正範囲を変更すれば項を変更することも可能である。しかしながら、表現パターンに偏性が見られ、表現パターンが少ないところが見受けられる。

これらの方法を使用すれば非二進探索アルゴリズムの ADC を実現することができる。これらは冗長ステップ数や表現パターンが多い出力レベルなど、性能がそれぞれで違うのでどの方法を利用すべきなのかは実際の SAR ADC 設計で異なると考えられる。

6. 電流源 DAC への応用

フィボナッチ数列を用いた非二進アルゴリズム SAR ADC の応用として電流源を用いた DAC を考える。

図 10 に二進探索アルゴリズムを用いた回路を示す。電流源 DAC はデジタル値でスイッチの on/off を制御して出力電流の値を決め、抵抗を通してアナログの電圧へ変換する。一般的には図 10 のように電流源の重み選択は、二進重みが利用され、回路面積が一番小さくなる。しかしながら個々の電流源には微小な電流のミスマッチが存在し、それらが組み合わさった出力電流のずれが出力電圧のずれとなり、DAC の静的な非線形性（単調性が保証されないこと）が問題となる。これを改善するために電流源の重みをすべて 1I の電流としたセグメント型 DAC が存在するが冗長性が大きく回路規模が大きい。

そこで適度な冗長性を持っているフィボナッチ数列を用いて、電流の重みを決める方法を考える。図 11 にその回路を示す。ADC が時間の冗長性を利用していたのに対して、この DAC は回路の冗長性（空間の冗長性）を利用する。主な応用は 3 つである。

- 1) この回路では、適度に冗長性を持つように電流源の大きさを定めているので、一つのデジタル入力に対して出力電流値の表現方法が複数個存在することになる。これを利用して、すべてのミスマッチが少なく一番線形性のよい組み合わせになるように電流源の選択ができる。

- 2) また別の利用法として、1 つのデジタルデータ値から複数の「電流源組み合わせ」の選択肢からランダムに使用する組み合わせを変更すること（ダイナミックマッチング）でミスマッチの影響を周波数拡散して SFDR（Spurious-Free Dynamic Range）の向上も期待できる。
- 3) さらにフィボナッチ数列の定義である

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

の下位 2 つの電流源の和が次の電流源になることを利用して、電流源を比較・校正できる。

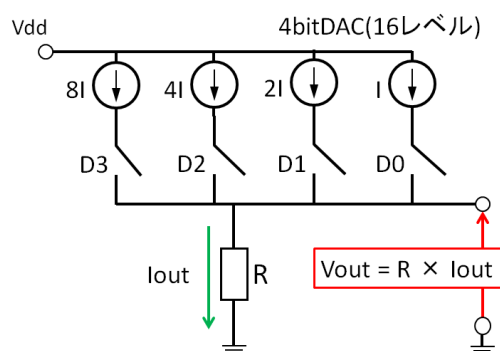


図 10. 二進重み電流源 DAC.

Fig.10 A binary-weighted current-steering DAC.

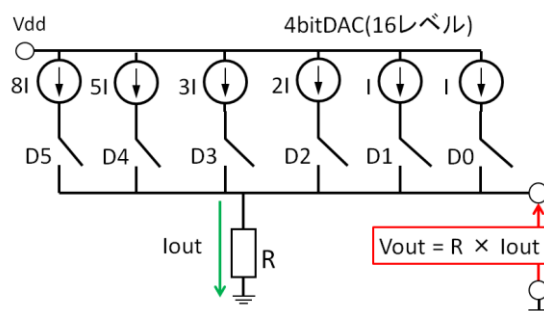


図 11. フィボナッチ重み電流源 DAC.

Fig.11 A Fibonacci-number weighted current-steering DAC.

7. まとめ

この論文では、逐次比較型 ADC において、非二進アルゴリズムを適用し冗長性を与えると誤動作に強くなること、さらには速度が上昇する可能性があることを考察した。その非二進探索アルゴリズムの基準電圧を決める方法はフィボナッチ数列の 1.6 進の性質を利用できることを示した。

フィボナッチ数列を用いた冗長性設計は電流源を用いる DAC にも応用することができ、自己校正やミスマッチを低減することができることを示した。

今後さらにフィボナッチ数列の発展として、前項の三項を足すトリボナッチ数列の利用を考えている。トリボナッチ数列は 1.8 進を実現でき、フィボナッチ数列と類似の性

質を持つために提示した内容と同様の考え方が適用でき、フィボナッチ数列よりも冗長性を少し下げた設計が可能になることが期待できる。

謝辞: 本研究をご支援いただいています半導体理工学研究センター(STARC)に感謝いたします。

文 献

- (1) 小林春夫:「デジタルアシスト・アナログ技術 - 微細CMOSの強みを生かす回路設計 -」大阪大学 高度人材育成センター 先端アナログ技術セミナー, 大阪 (2012年1月).
- (2) 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).
- (3) 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).
- (4) 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).

付録. フィボナッチ数列の性質

フィボナッチ数列には黄金比以外にも、さまざまな性質がある。それらは今後の研究に応用できる可能性があるので付録としてここに示す[4]。以下の性質はすべてのフィボナッチ数で必ず成り立つものである。ここで n は $n \geq 1$ となる任意の自然数である。

- ①連続する 10 個のフィボナッチ数の和は 11 で割り切れる。(A|B: B は A で割り切れる)
- 11 | ($F_n + F_{n+1} + F_{n+2} + F_{n+3} + F_{n+4} + F_{n+5} + F_{n+6} + F_{n+7} + F_{n+8} + F_{n+9}$)
- ②連続するフィボナッチ数は互いに素である。つまり、両者の最大公約数は 1 である。
- ③合成数番目のフィボナッチ数(4番を除く)も合成数である(合成数=素数でない数)。これを別の言い方で表すと n が素数でない場合、 F_n は素数ではない。
- ④フィボナッチ数の最初の n 個の和は 2 つ後の項から 1 引いたものに等しい。

$$\sum_{i=1}^n F_i = F_1 + F_2 + F_3 + \dots + F_n = F_{n+2} - 1$$

- ⑤連続する偶数番のフィボナッチ数の和は、和の最後の偶数番のフィボナッチ数の次のフィボナッチ数より 1 小さい。

$$\sum_{i=1}^n F_{2i} = F_2 + F_4 + F_6 + \dots + F_{2n-2} + F_{2n} = F_{2n+1} - 1$$

- ⑥連続する奇数番のフィボナッチ数の和は、和の最後の奇数番のフィボナッチ数の次のフィボナッチ数に等しい。

$$\sum_{i=1}^n F_{2i-1} = F_1 + F_3 + F_5 + \dots + F_{2n-1} = F_{2n}$$

- ⑦フィボナッチ数の平方の和は、最後の数とその次のフィボナッチ数との積に等しい(黄金らせんを描く)。

$$\sum_{i=1}^n F_i^2 = F_n F_{n+1}$$

- ⑧2つの交互的フィボナッチ数の平方の差は、両者の番号の和を番号とするフィボナッチ数に等しい。

$$F_n^2 - F_{n-2}^2 = F_{2n-2}$$

- ⑨2つの連続するフィボナッチ数の平方の和は、その番号の和を番号とするフィボナッチ数に等しい。

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

- ⑩4つの連続するフィボナッチ数については、中2項の平方の差が両端の2項の積に等しい。

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

- ⑪交互的フィボナッチ数の2つの積は、両者の間にあるフィボナッチ数の平方より1多いか少ないか、いずれかである。

$$F_{n-1} F_{n+1} = F_n^2 + (-1)^n$$

- ⑫選んだフィボナッチ数の平方とそのフィボナッチ数から等距離にあるフィボナッチ数の積の差は、別のフィボナッチ数の平方である。(ただし $k \geq 1$)

$$F_{n-k} F_{n+k} - F_n^2 = \pm F_k^2$$

- ⑬ mn 番目のフィボナッチ数 F_{mn} は、 m 番目のフィボナッチ数 F_m で割り切れる。

- ⑭連続するフィボナッチ数の積の和は、フィボナッチ数の平方に等しいか、フィボナッチ数の平方より1小さい。

$$n \text{ が奇数のとき } \sum_{i=2}^{n+1} F_i F_{i-1} = F_{n+1}^2$$

$$n \text{ が偶数のとき } \sum_{i=2}^{n+1} F_i F_{i-1} = F_{n+1}^2 - 1$$

- ⑮黄金比と黄金比の逆数の差は丁度1である。

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

$$\lim_{n \rightarrow \infty} \frac{F_{n-1}}{F_n} = 0.618033988749895 = \frac{1}{\phi}$$

すなわち以下の方程式が成り立つ、唯一の正の値が黄金比である。

$$1/\phi = \phi - 1$$

- ⑯黄金比 ϕ のべき乗は以下の方程式に従い、 a と b は必ずフィボナッチ数である。

$$\phi^n = a\phi + b$$