

フィボナッチ冗長設計逐次比較近似 AD変換器の補正力の
定量化及びフィボナッチ重み付けDACの提案

群馬大学大学院
電子情報部門
荒船拓也



アウトライン

- 研究背景
- 逐次比較近似AD変換器の冗長設計
- フィボナッチ冗長SARADC
- フィボナッチ冗長設計の実現性
- まとめ

アウトライン

- 研究背景
- 逐次比較近似AD変換器の冗長設計
- フィボナッチ冗長SARADC
- フィボナッチ冗長設計の実現性
- まとめ

研究背景・目的

自動車のエレクトロニクス化が著しく
車載用エレクトロニクス技術に大きな関心



車載用マイコンと組み合わせるADCへの要求が厳しい

➡ 高性能化が必要

↓ +冗長性

逐次比較近似AD変換器の冗長設計

➡ 最適な設計手法が確立されていない



↓ +整数論

研究目的

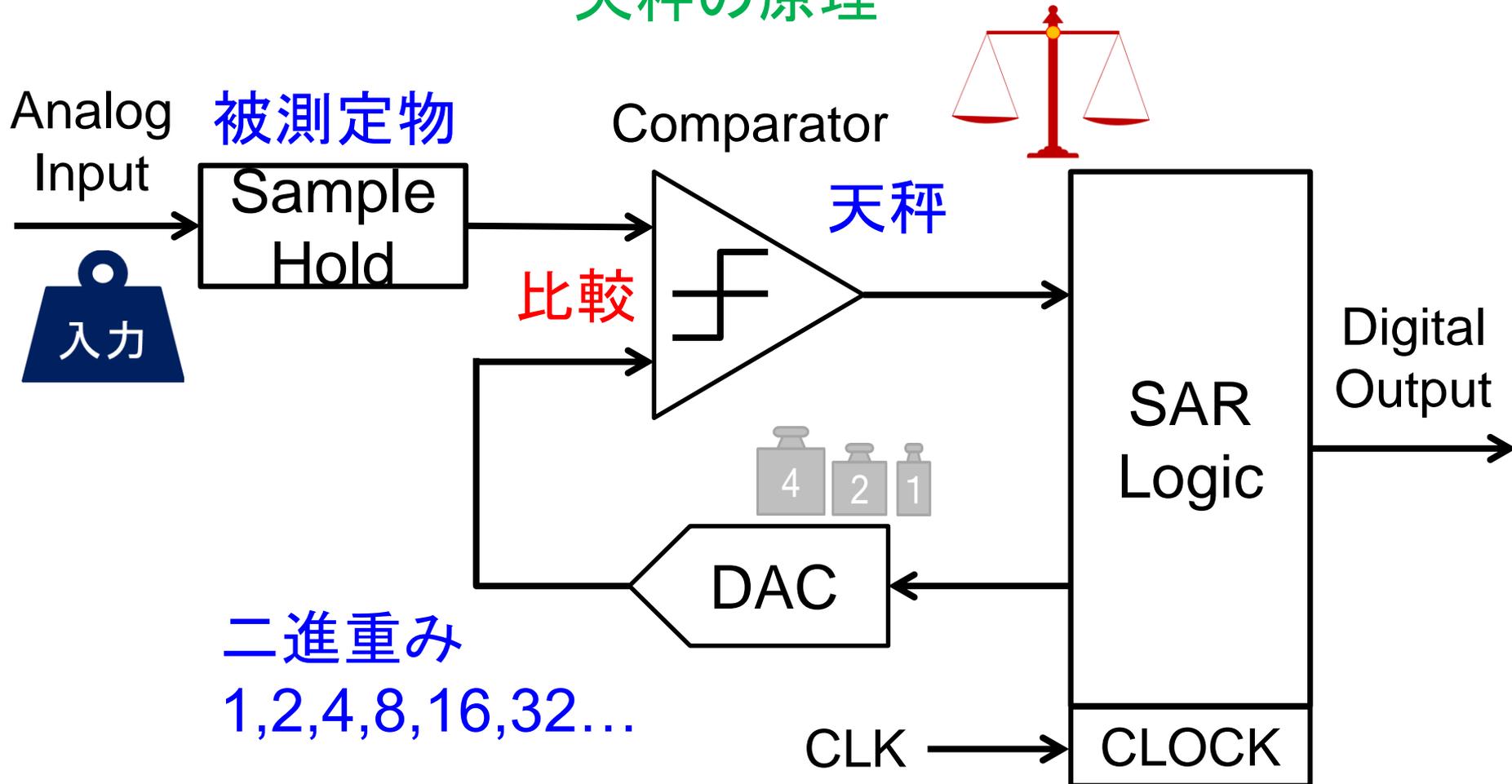
逐次比較近似AD変換器の整数論を用いた冗長設計

➡ 整数論の性質を用いて簡単に実現

逐次比較近似AD変換器の構成

アナログ入力と重み電圧を比較、結果に応じたデジタル出力

天秤の原理



二進探索SAR ADCの正動作

5bit-5step SAR ADC

$$X + 4 + 1 + 0.5 + 0.5 = 16 + 8 + 2$$



$$X = 16 + 8 - 4 + 2 - 1 - 0.5 - 0.5 \Rightarrow 20$$

十進数と二進数が一対一に対応

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	1	0	1	0	0	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

$$16 + 8 - 4 + 2 - 1 - 0.5 - 0.5 \Rightarrow 20$$

1 0 1 0 0

20

Level

二進探索SAR ADCの誤動作

5bit-5step SAR ADC

$$X + 4 + 2 + 0.5 = 16 + 8 + 1 + 0.5$$



$$X = 16 + 8 - 4 - 2 + 1 + 0.5 - 0.5 \Rightarrow 19$$

十進数と二進数が一対一に対応

一回の誤判定が出力ミスに
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						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

判定誤り

1 0 0 1 1

16 + 8 - 4 - 2 + 1 + 0.5 - 0.5 ⇒ 19

アウトライン

- 研究背景
- 逐次比較近似AD変換器の冗長設計
- フィボナッチ冗長SARADC
- フィボナッチ冗長設計の実現性
- まとめ

冗長性を有するSAR ADC

冗長：余裕、予備



SARADCへ適用

時間的冗長性の利用

- 判定回数の増加
- 非二進重み

デジタルコード表現が複数
となる範囲 $q(k)$ 発生

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

$q(k)$: k ステップ目の補正能力

冗長性を有するSAR ADC

冗長：余裕、予備



時間的冗長性の利用

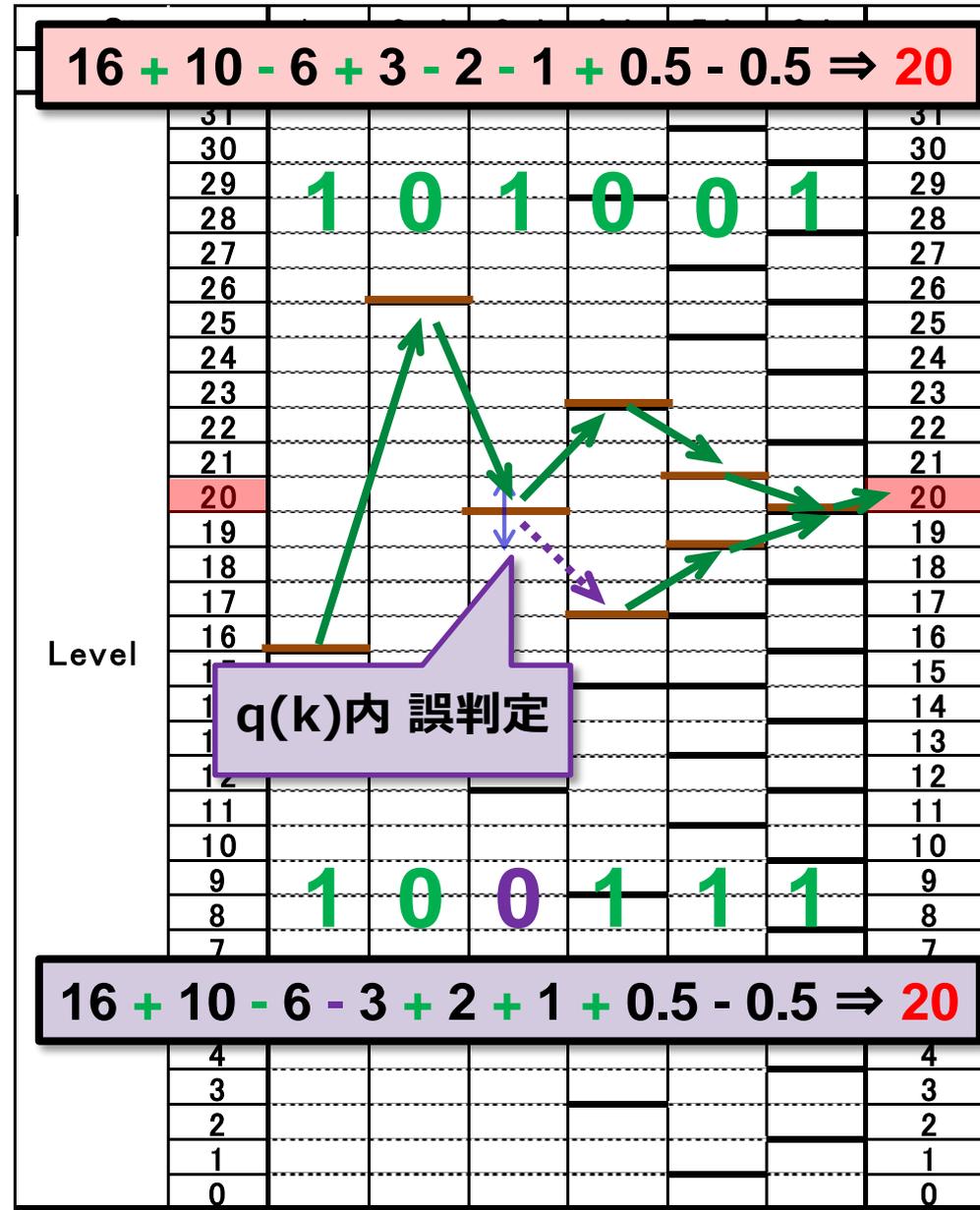
- 判定回数の増加
- 非二進重み

デジタルコード表現が複数
となる範囲 $q(k)$ 発生

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



冗長設計における
補正力は $p(k)$ で決まる



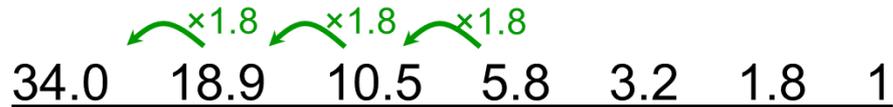
比較電圧重み $p(k)$ の決定(従来手法)

従来手法

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

- 適切な基数の決定が難しい
- $p(k)$ は必ず小数になる(単位項による実現困難)

例 : Radix1.8 Weight



② 条件を定めて総当たり法

- 全パターン検討に膨大な時間がかかる
- 最適なパターン検出が難しい
- 条件の小さな変化に対応しづらい

③ 最も適当な重みを補正力 $q(k)$ で決定する

- 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

アウトライン

- 研究背景
- 逐次比較近似AD変換器の冗長設計
- **フィボナッチ冗長SARADC**
- フィボナッチ冗長設計の実現性
- まとめ

フィボナッチ数列の応用

フィボナッチ数列

定義 (n=0,1,2,3...)

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

$$F_0 = 0, F_1 = 1$$

性質

隣り合う項の比率は“**黄金比**”に収束

$$\lim_{n \rightarrow \infty} \frac{F_n}{F_{n-1}} = 1.6180339887 \dots$$

2点の性質を発見!

- ① 全ての入力値を補正できる
⇒ エラー耐性の高い変換ができる
- ② 隣接stepでの補正範囲が必ず接する
⇒ 速いAD変換ができる [1]

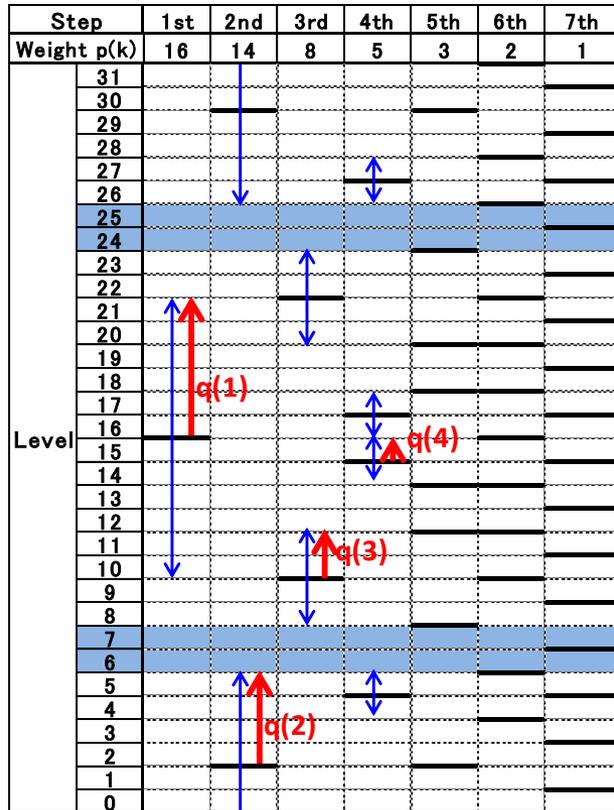
Step	1st	2nd	3rd	4th	5th	6th	7th
Weight p(k)	16	8	5	3	2	1	1
Level							
31				↓			
30			↑		↑		
29					↑		
28			↓		↑		
27					↑		
26		↑		↑			
25				↓			
24					↑		
23					↑		
22				↑			
21		↓		↓			
20	↑		↑		↑		
19					↓		
18					↑		
17			↓		↓		
16				↑			
15				↓			
14			↑		↑		
13					↑		
12					↑		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

[1] Yutaro Kobayashi, Haruo Kobayashi, "SAR ADC Algorithm with Redundancy Based on Fibonacci Sequence", The Joint Conference 4S-2014/AVIC 2014, Ho Chi Minh City, Vietnam (Oct. 22-24, 2014)

従来手法との比較(5bit ADC)

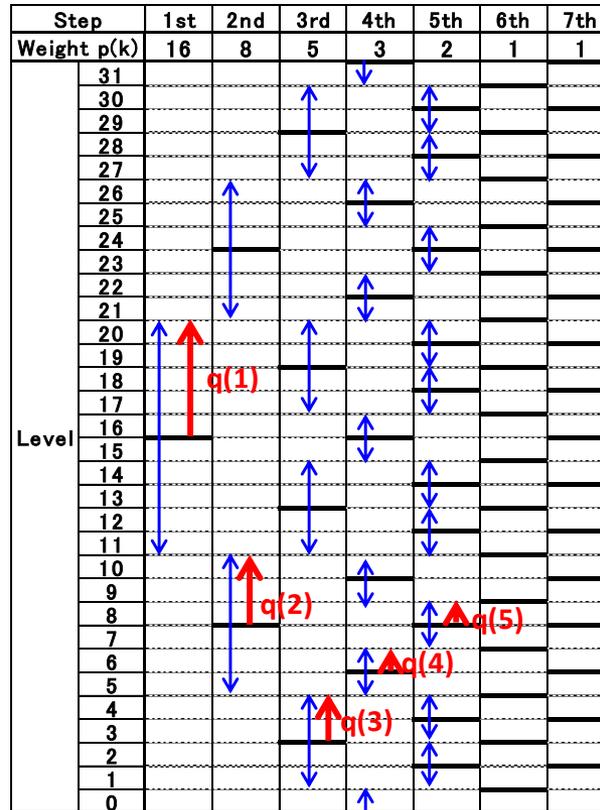
従来手法

1.70進数



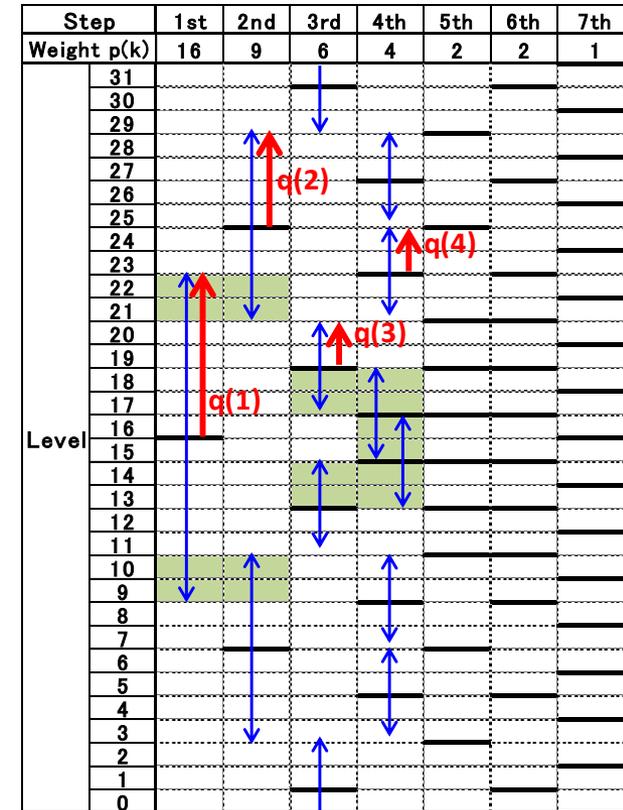
フィボナッチ手法

1.62進数



従来手法

1.55進数



フィボナッチ数列冗長手法

エラー耐性が強い
変換速度が速い

アウトライン

- 研究背景
- 逐次比較近似AD変換器の冗長設計
- フィボナッチ冗長SARADC
- **フィボナッチ冗長設計の実現性**
- まとめ

冗長回路実現の大きな障壁

従来手法

SARロジック回路とDACの変更により実現

手法1:フィボナッチ加減算SARロジック

- SARロジックが二進よりも回路が大規模化・複雑化・遅延
- フィボナッチ加減算を回路上で実現することが困難

手法2:冗長SARADC用のSARロジック

- RAMや減算器、温度計デコーダ等が必要
- 二進数に一度エンコードし、演算後デコードが必要
- 比較電圧重みの事前記録が必要

本提案手法

DACの変更のみで実現

- 従来のSARロジックを使用可
- 従来DACの改良によりフィボナッチ重み付けできる
- 二進SARADCとほぼ同面積・同構造で実現可能

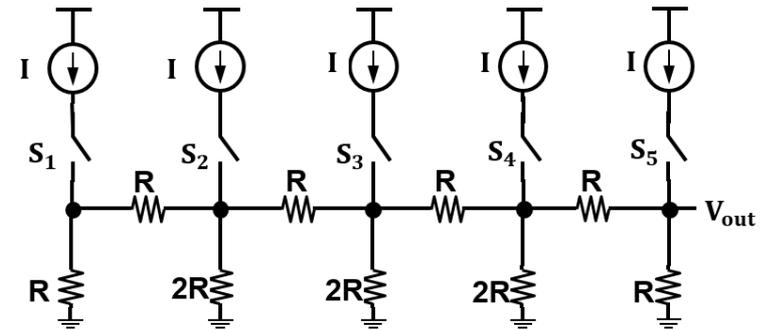
フィボナッチDA変換器の新提案回路

新しい発見！

従来DAC：R-2R抵抗ラダー回路

- 2進重みの電圧発生

↓ 抵抗値を全てRで統一

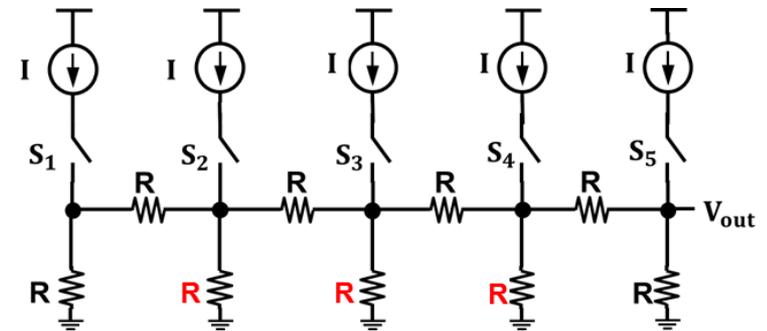


R-2R抵抗ラダー回路

提案DAC：R-R抵抗ラダー回路

- フィボナッチ重みの電圧発生

↓



R-R抵抗ラダー回路

簡単な構成でフィボナッチ
対応DA変換器実現可！

R-Rラダー (フィボナッチDACの原理)

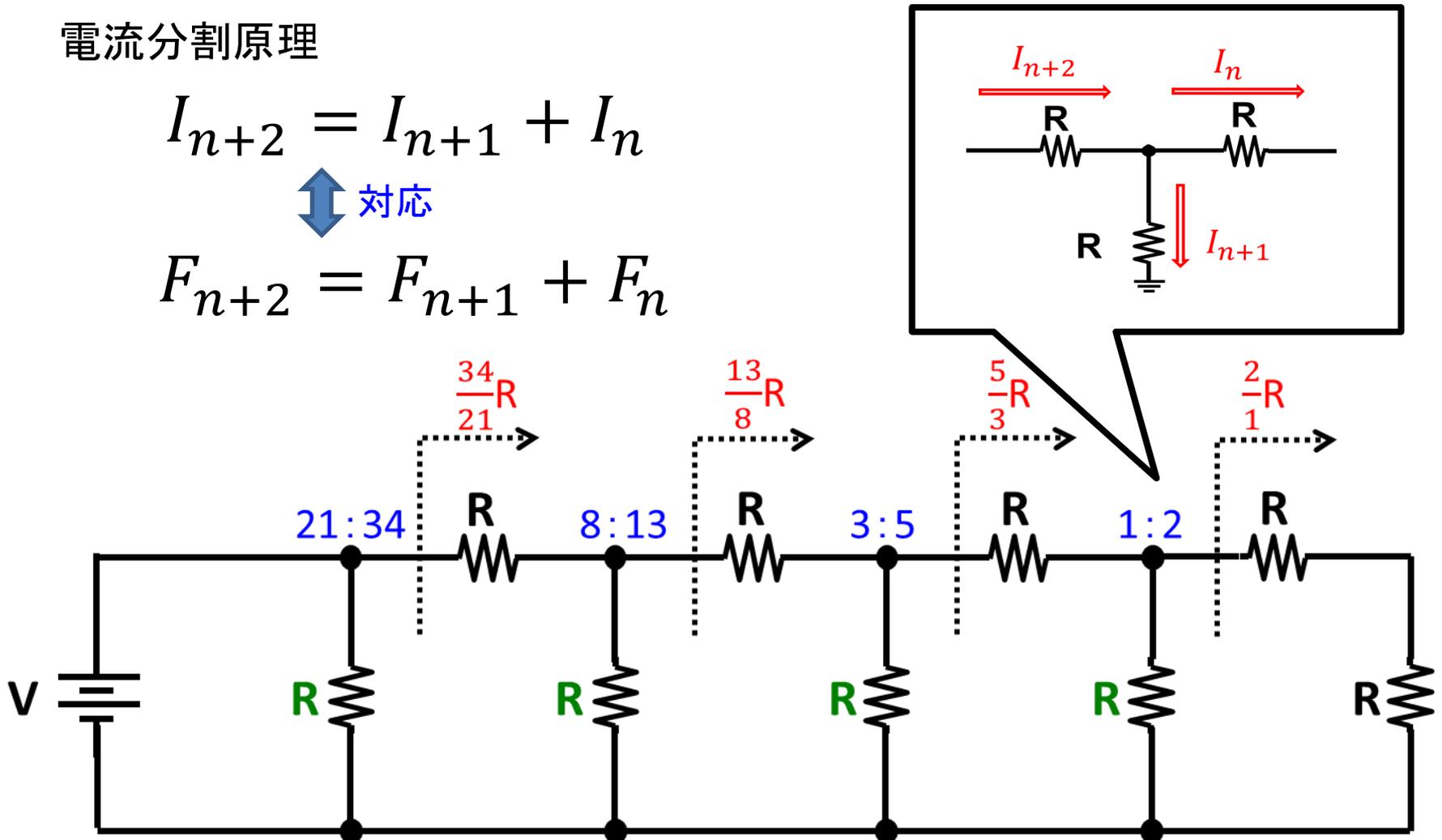
R-Rラダー回路 ➤ ノード毎で電流をフィボナッチ比(黄金比)に分割！

電流分割原理

$$I_{n+2} = I_{n+1} + I_n$$

↕ 対応

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



2つの逐次探索方法

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

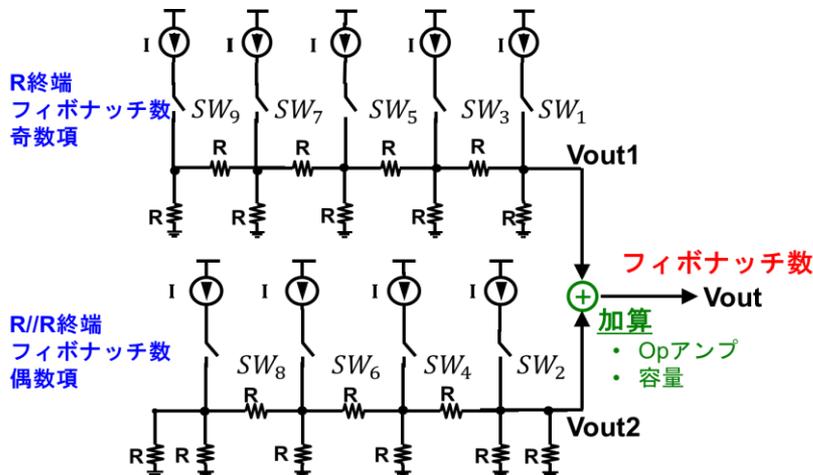
加算型フィボナッチDAC

値を**加算**するか**加算しないか**を判定

- ・二進SARロジックが**使用不可**
- ・抵抗ラダー列が**2段必要**
- ・加算回路が余分に**必要**
- ・補正力を十分に**発揮できない**

加減算型フィボナッチDAC

値を**加算**するか**減算**するかを判定



加算型R-R抵抗ラダー回路

フィボナッチ減算性質を応用した新たな可能性

定義

$$F_0 = 0$$

$$F_1 = 1$$

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

初項から計算していくと...

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233...

性質

数列の奇数項を偶数項の減算で表現可能

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

➡ **偶数項のみで全てのフィボナッチ重み生成可**



Leonardo Fibonacci
(伊:1170~1250年頃)

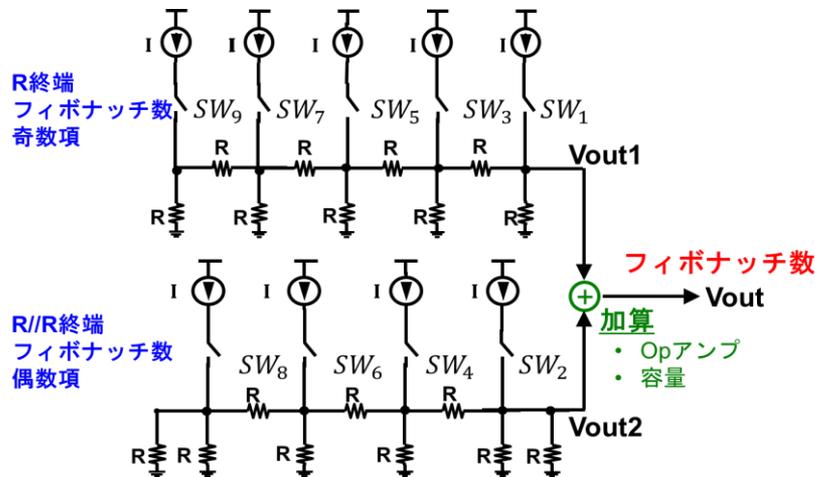
2つの逐次探索方法

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

加算型フィボナッチDAC

値を**加算**するか**加算しないか**を判定

- ・二進SARロジックが**使用不可**
- ・抵抗ラダー列が**2段必要**
- ・加算回路が余分に**必要**
- ・補正力を十分に**発揮できない**

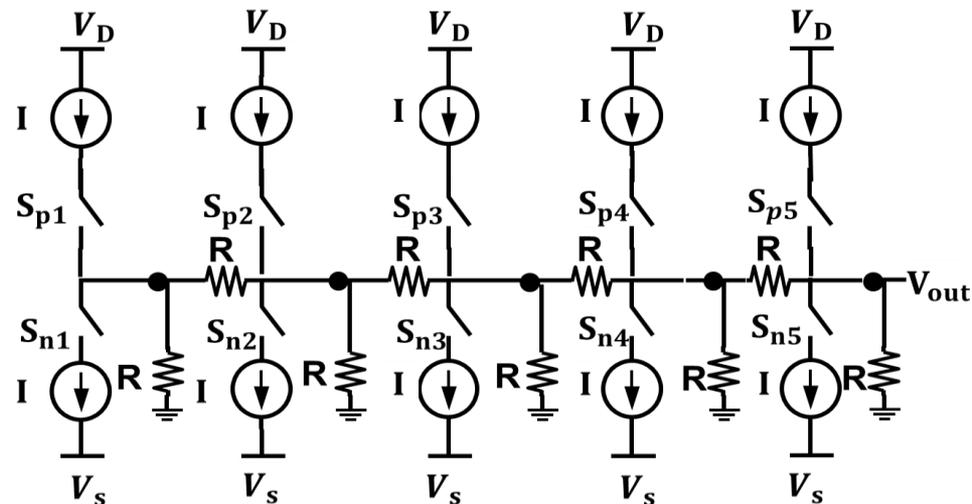


加算型R-R抵抗ラダー回路

加減算型フィボナッチDAC

値を**加算**するか**減算**するかを判定

- ・二進SARロジックを**使用可能**
- ・抵抗ラダー列を**1段に**
- ・加算部が**不要**
- ・補正力を**発揮できる**



加減算型R-R抵抗ラダー回路

シミュレーション条件

提案DACを設計し動作確認

◆詳細

5bit 加減算型R-R抵抗ラダーDAC

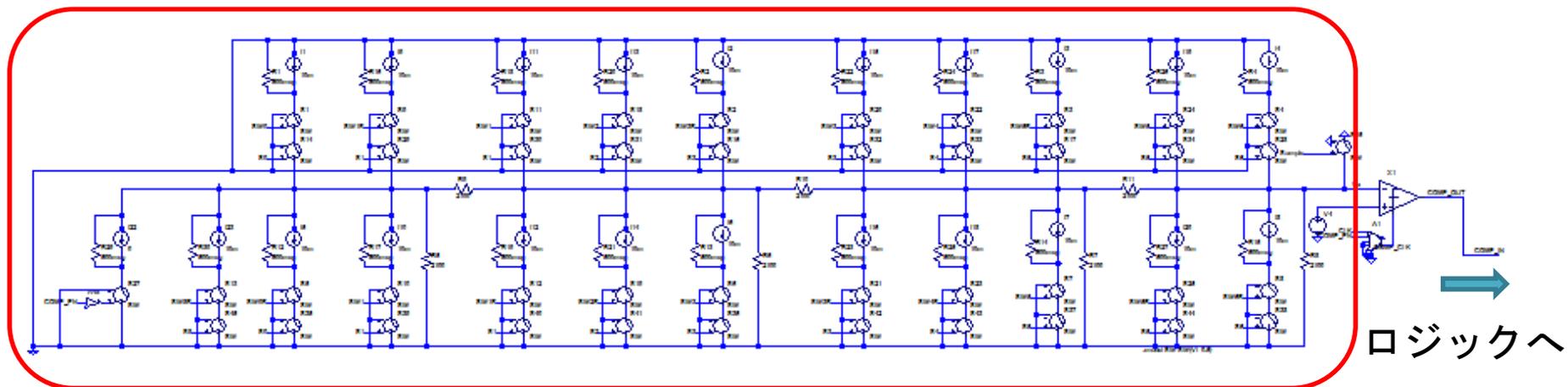
コンパレータ (Behavior Modelを利用した理想コンパレータ)

出力デジタル値のフルスケールは1Vとして

5bitDACのフルスケールは $31V[2^5 - 1]$

駆動クロック周期 $1.1111\mu s$

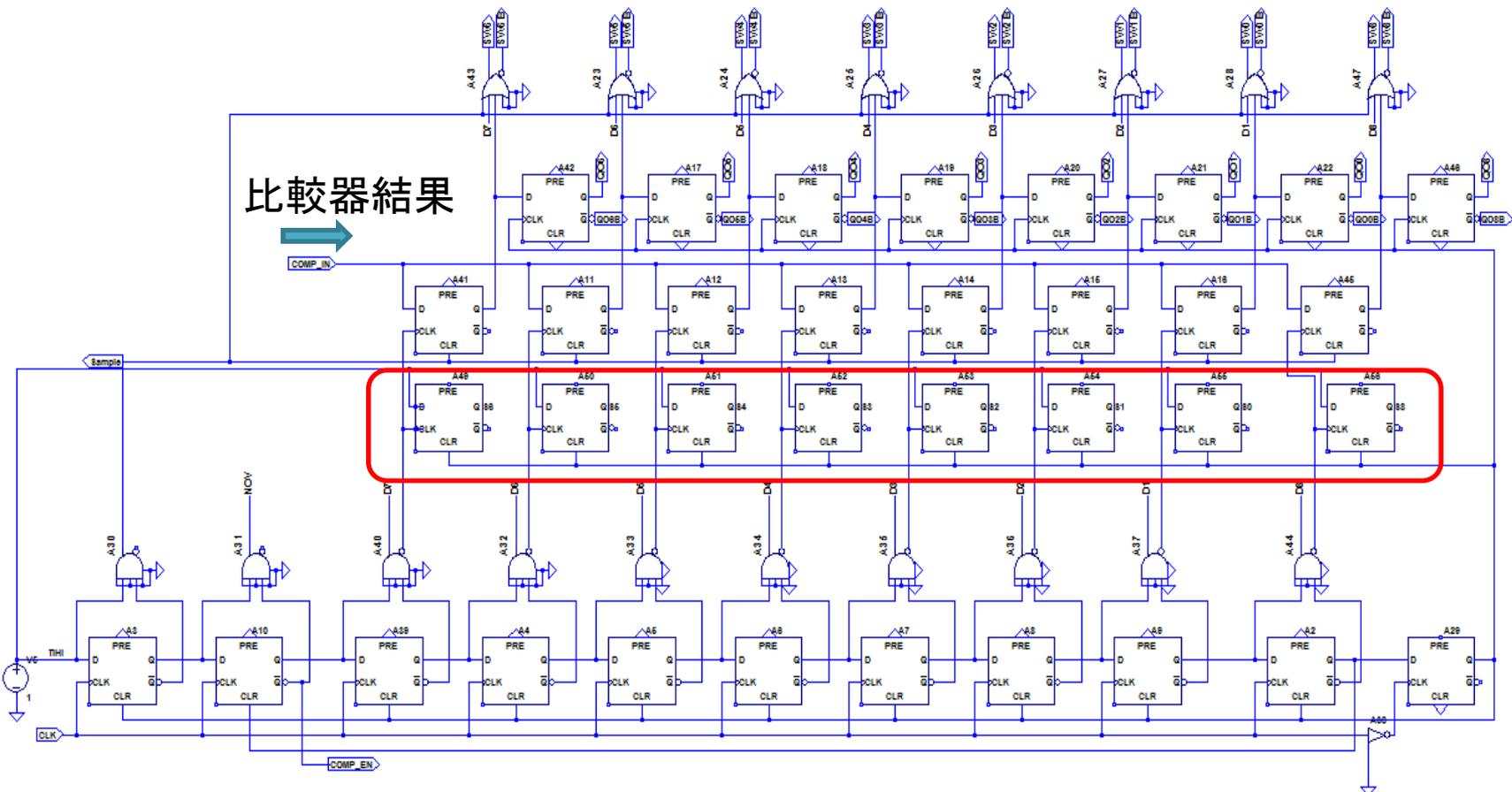
入力信号はS&H処理後のデジタル信号



SARロジック

5bit-SAR Logic

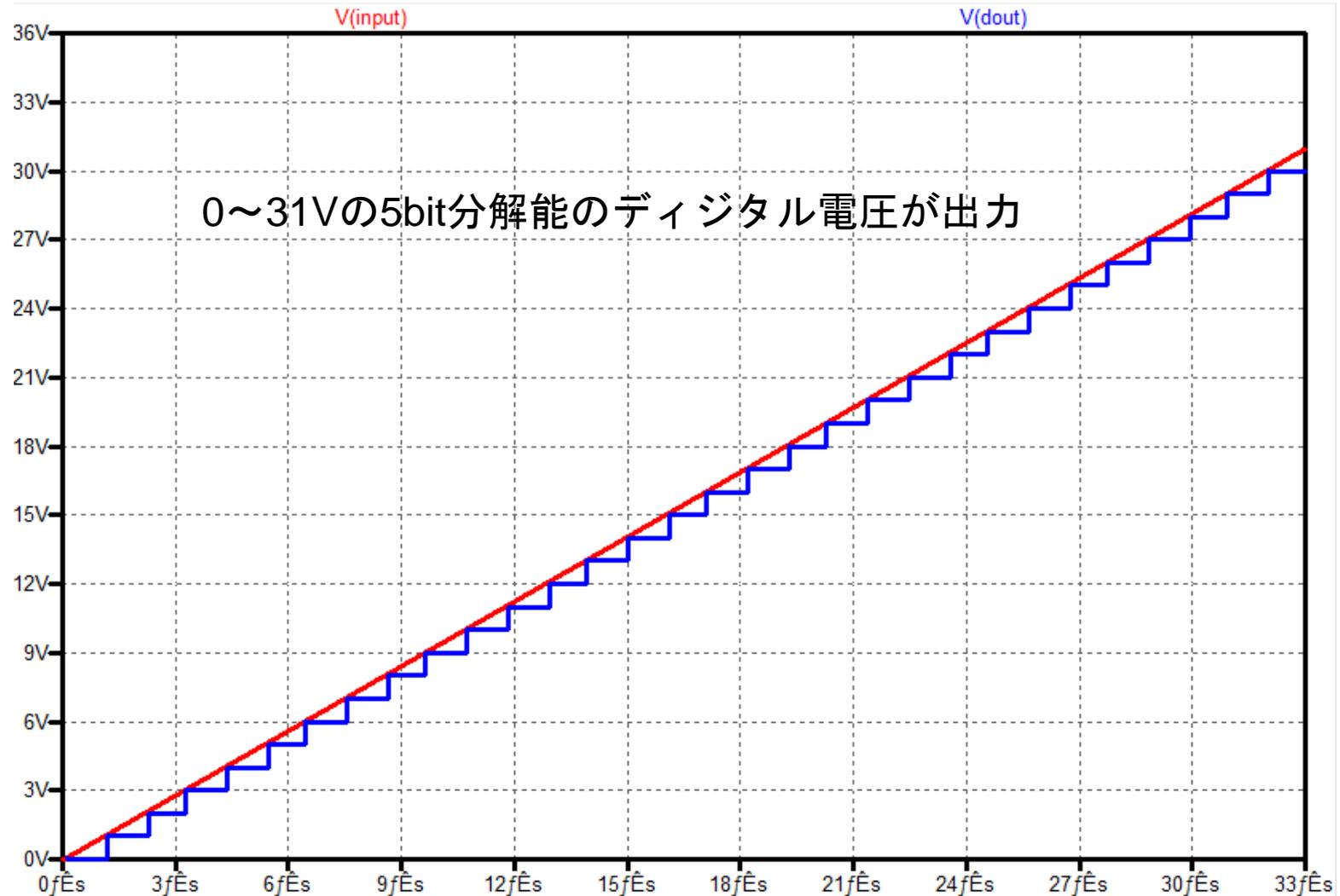
- 二進のSARロジックを使用
- **初期動作制御部**としてD-FF段を1段だけ追加



シミュレーション結果

正常なAD変換動作

- : 入力アナログ電圧
— : 出力電圧



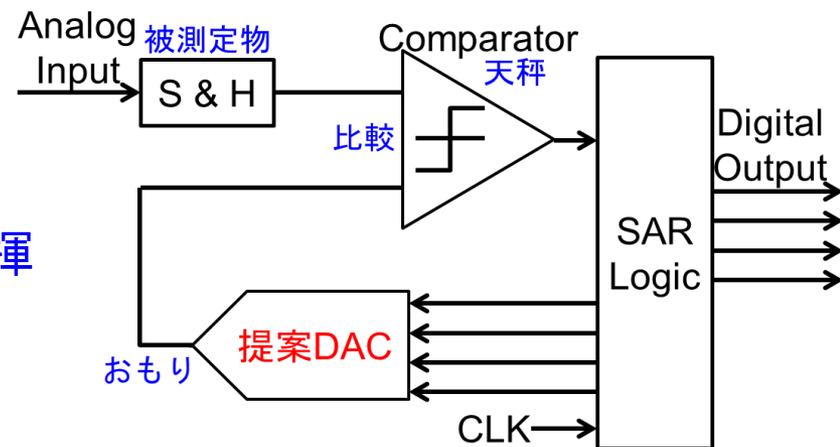
アウトライン

- 研究背景
- 逐次比較近似AD変換器の冗長設計
- フィボナッチ冗長SARADC
- フィボナッチ冗長設計の実現性
- まとめ

まとめ

提案回路の性質

- DACでフィボナッチの加減算が可能
 - フィボナッチ設計の能力を最大限に発揮
- 加算と減算が1対1で対応
 - 従来SARロジックをそのまま使用可能
- 偶数項減算による奇数項表現が可能に
 - ラダー列を1列で全てのフィボナッチ重み電圧を表現



提案回路の可能性

- 他の冗長設計よりも小規模回路で実現可能
- 提案DACを二進SARADCに適用するだけでフィボナッチSARADCを実現

補正能力・変換速度に優れたAD変換器を容易に実現可能

整数論の回路応用

「世界は数理でできている。」

アイザック・ニュートン



Isaac Newton
(英:1642-1727)

AD変換器
DA変換器

+

整数論
(数理)

=

新たな
可能性

整数論は様々な分野に適用することで
無限大の可能性が秘められている

「回路は数理で設計できる」

T. Arafune

Q&A

Q. 今回の成果はフィボナッチ分割を発見したこと？

A. それ以外にもフィボナッチ減算性質を用いることでラダー列を1段にすることができたことが回路的には大きな成果であると考えています。

Q. 自動車等に応用する際に問題になることはあるのだろうか？

A. 従来のDACを使用しているため大きな問題はないと思います。しかし、実際の自動車に使用されているDACが今回提案した方式とは限りませんので、それに応じて回路変更を行う必要があると思います。

付録

シミュレーション条件

Radixとフィボナッチ手法の補正力を比較

◆ 入力値

刻み幅0.1V

フルスケールは63V

S&H処理後の入力を想定

◆ ノイズ量

DACのフルスケールの±5%(±3V)

刻み幅0.5V

6ステップの内1ステップにノイズを与える

フルレンジを超えた入力は判定数に含めない

(入力31Vに0.5Vのノイズ成分を重畳させた場合)

◆ 調査した重み付け

① 基数radixから決定する(radix刻み幅0.01)

隣り合う項の比率を1~2の区間で決定

毎回同じ比率を利用

Radixの刻み幅は0.01

② フィボナッチ重みを適応

初2項は1,1から初めフィボナッチ数列式に基づき重み付けを行う

外部ノイズが瞬間的に(1step幅)混入した場合を想定
ノイズ耐性は総判定数に対する誤判定数の割合で示す

$$\text{ノイズ耐性} = \frac{\text{誤判定数}}{\text{総判定数}}$$

radix1.70手法

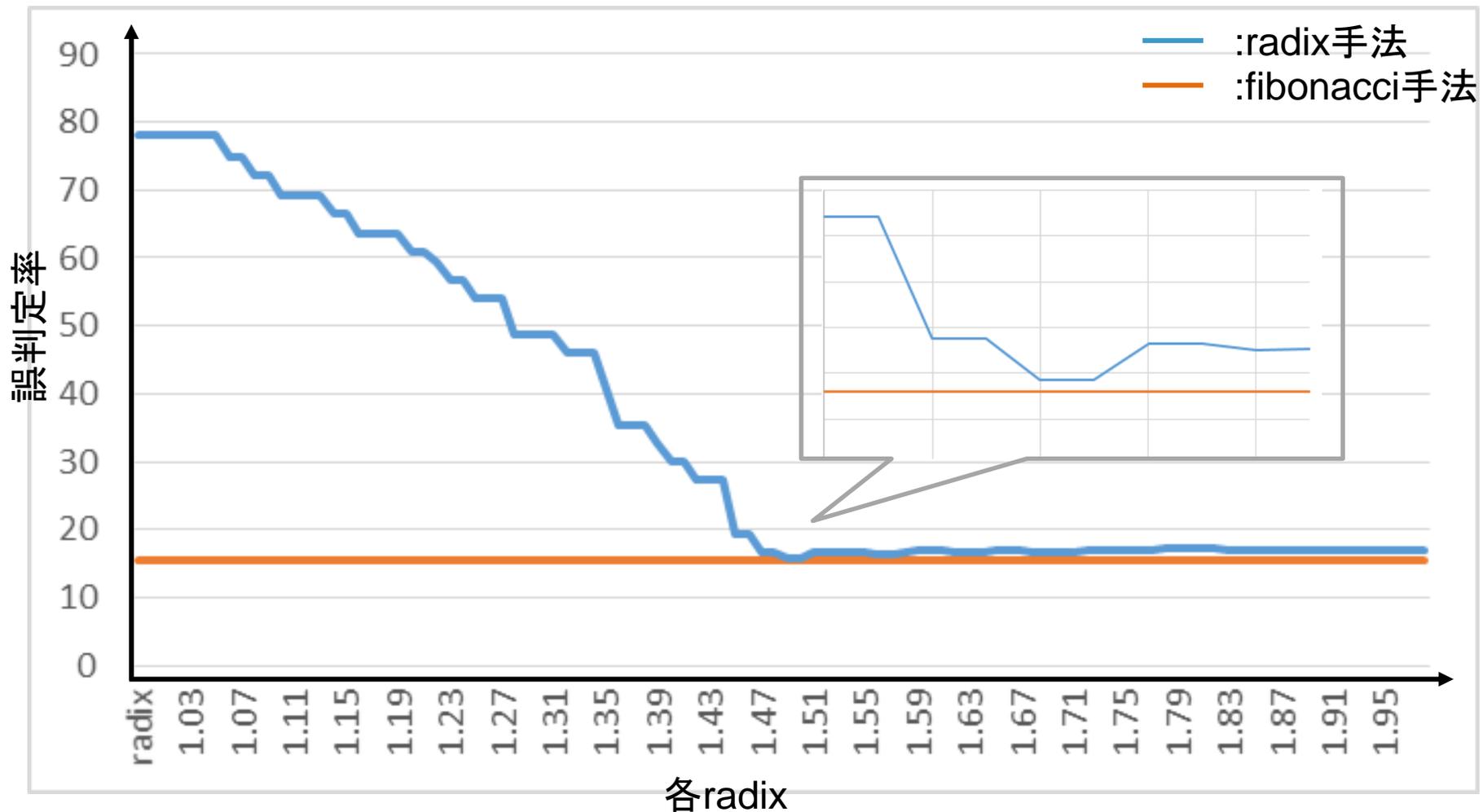
Step	1st	2nd	3rd
Weight p(k)	16	8	5
31			
30			
29			
28			
27			
Level	23		
22			
21			
20			
19			
18			
17			
16			
15			

入力値 : 23.8V
ノイズ段 : 2nd-step
ノイズ量 : +0.5V

noise



補正力の結果



フィボナッチ手法はradix手法と比較して誤判定が最も少ない

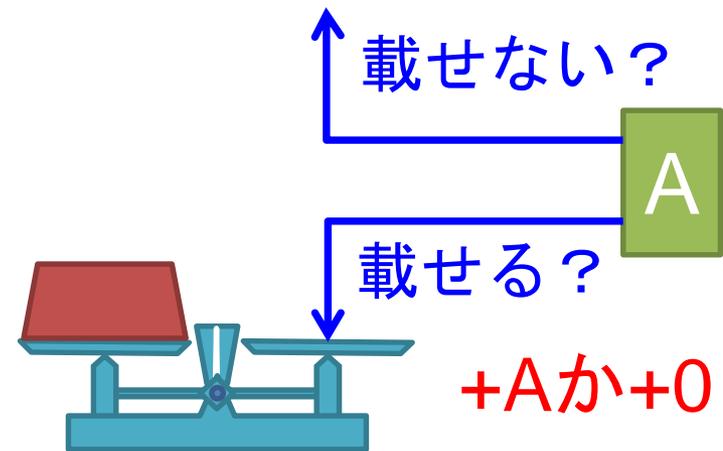
※便宜上、フィボナッチ手法の結果は全Radixに表示

2つの逐次探索方法

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

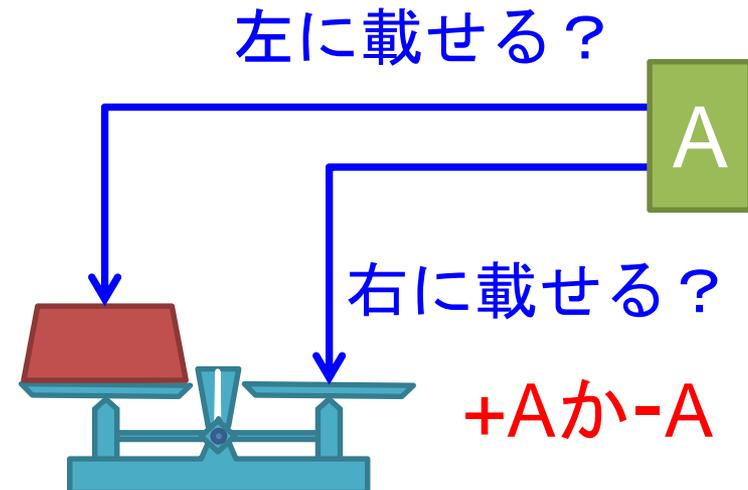
加算型SARロジック

値を**加算**するか**加算しない**かを
決める方法



加減算型SARロジック

値を**加算**するか**減算**するか
を決める方法

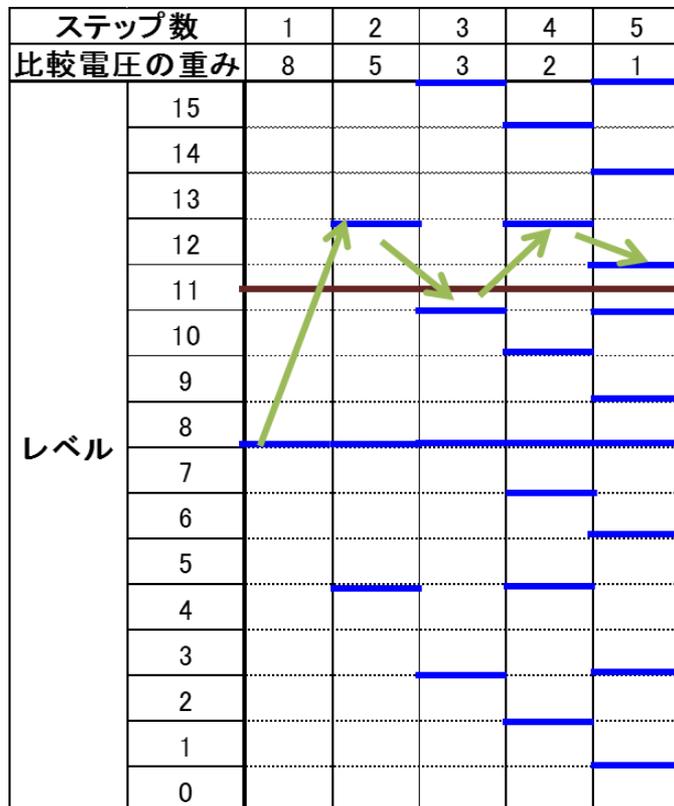


2つの逐次探索方法

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

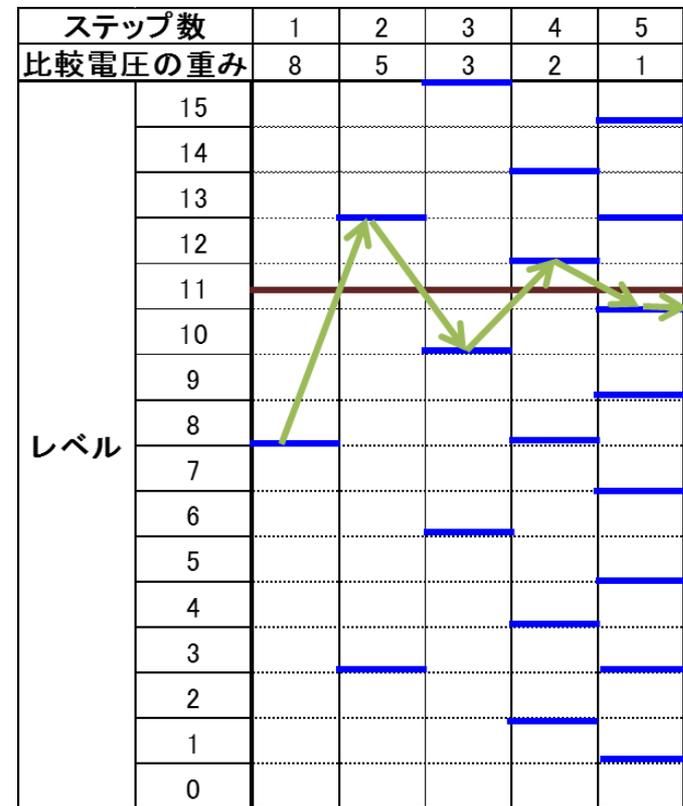
加算型フィボナッチDAC

冗長設計による補正力が
発揮できない

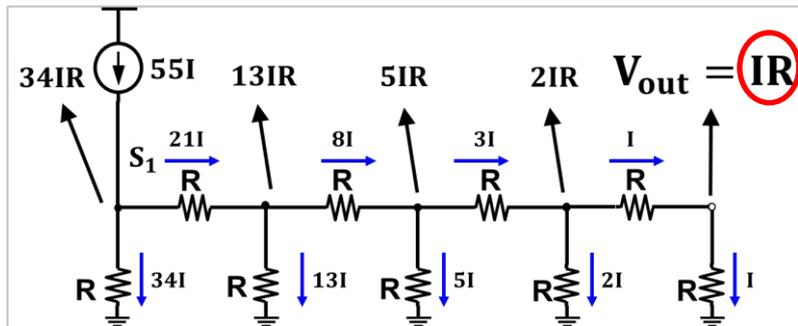


加減算型フィボナッチDAC

冗長設計による補正力を
発揮できる



R終端回路の動作

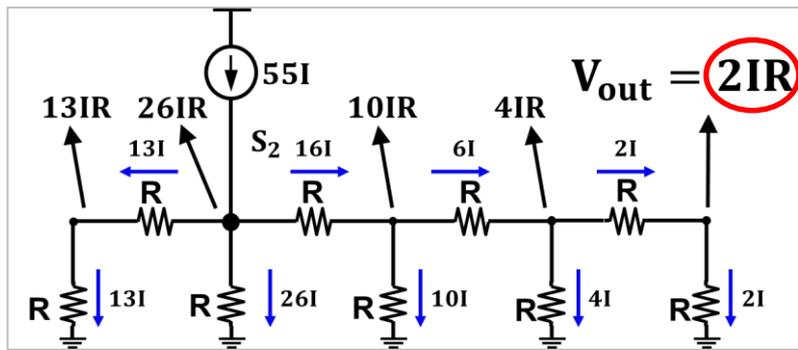


N点ノードの回路を考えると...

右からm点目に電流源を入れたときの出力電圧 V_{out} は

$$V_{out}(m) = \left(\frac{F_{2(N-m)+1}}{F_{2N}} \right) IR$$

→ 奇数項の出力ができる

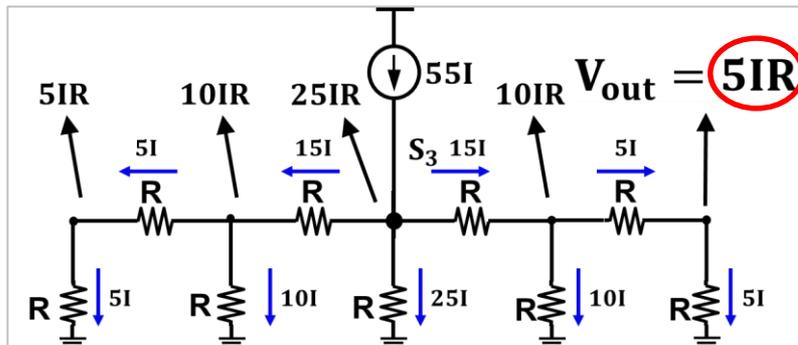


左図のように5点ノードの回路を考えると...

フィボナッチ数列

1, 1, 2, 3, 5, 8, ...

奇数項の出力



R||R 終端回路の動作

N点ノードの回路を考えると...

右からm点目に電流源を入れたときの出力電圧Voutは

$$V_{out}(m) = \left(\frac{F_{2(N-m+1)}}{F_{2(N+1)}} \right) IR$$

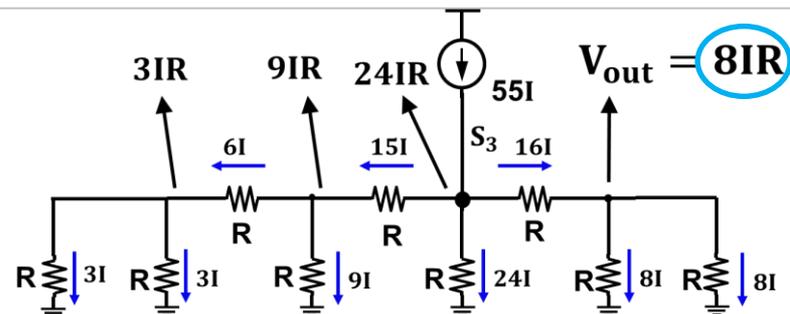
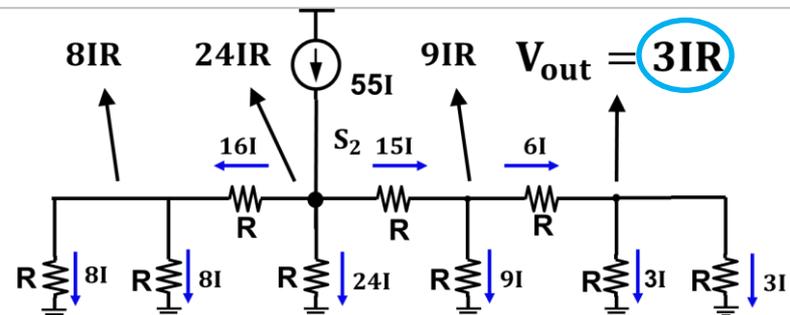
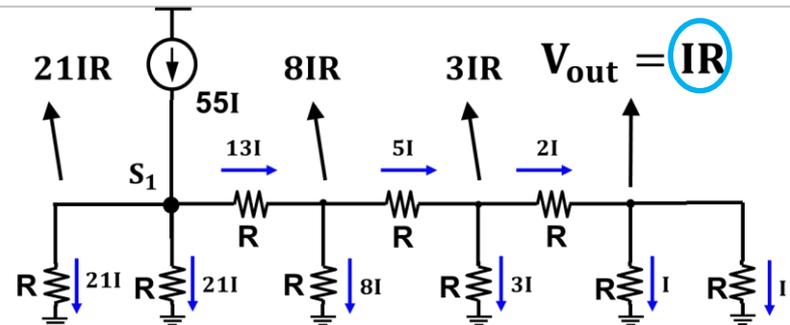
➡ 偶数項も出力可

左図のように4点ノードの回路を考えると...

フィボナッチ数列

1, 1, 2, 3, 5, 8, ...

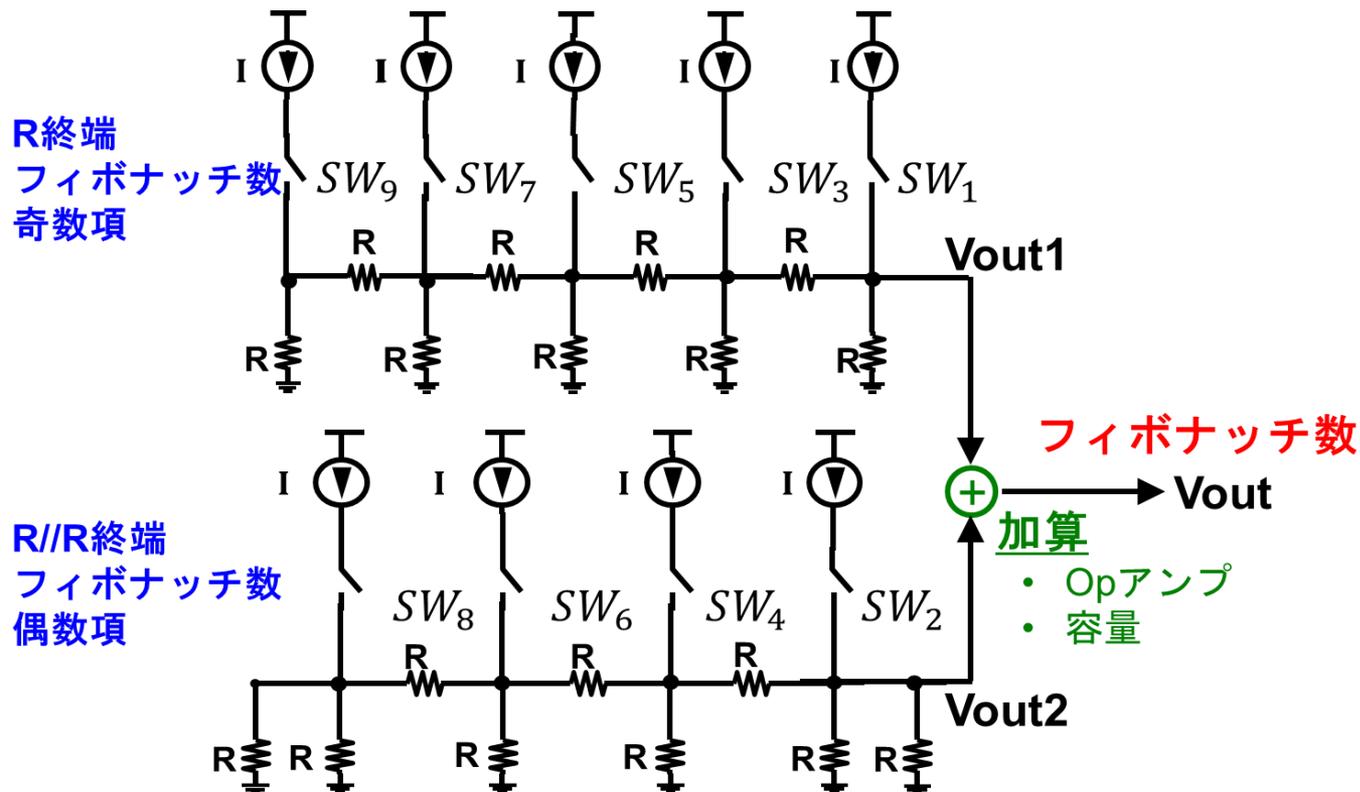
偶数項の出力



従来のフィボナッチ重み付けR-RラダーDAC

回路の大規模化

- 原理的にR-Rラダー回路が2列必要
- 加算部が必須



加減算手法(奇数項)

従来方式

加算用の電流源のみ



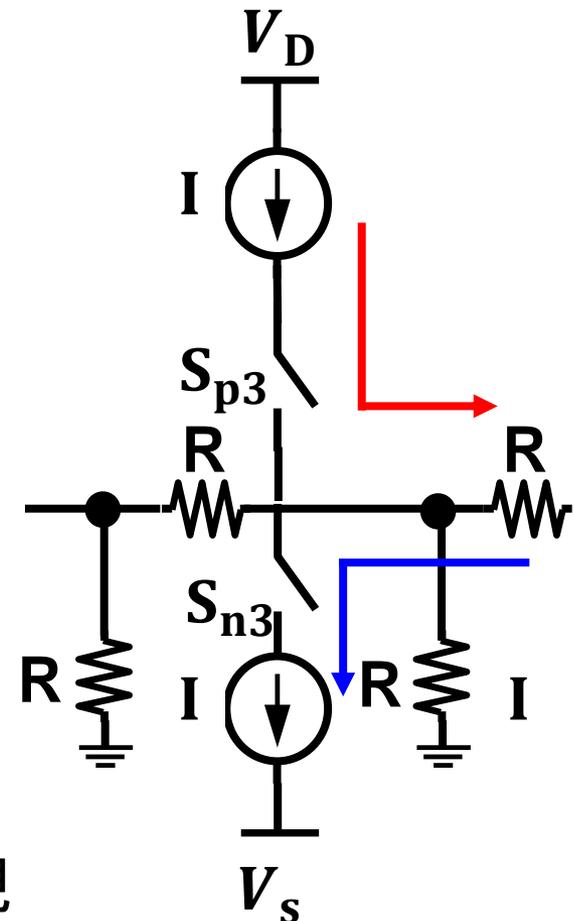
提案方式

加算用と減算用の電流源を
各ノードに1対1で用意

加減算原理

回路全体に流れる電流の加減算で
フィボナッチ重みの加減算を等価的に実現

加算用電流源



減算用電流源

スイッチの制御

加算時

加算スイッチ S_p をON



電流を加算し
出力ノードの電流増加

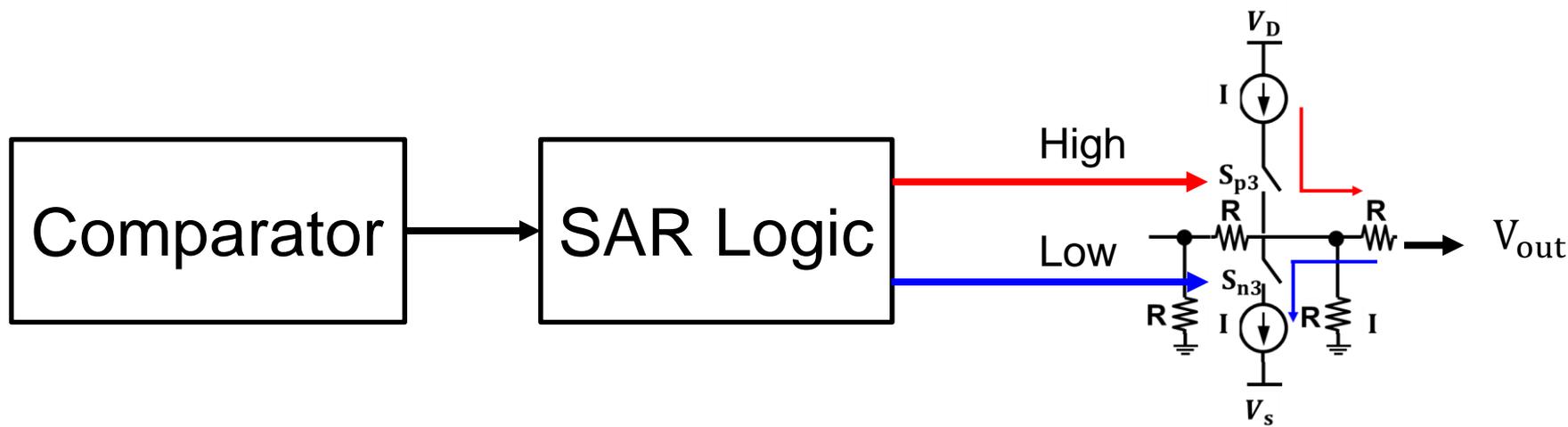
減算時

減算スイッチ S_n をOFF



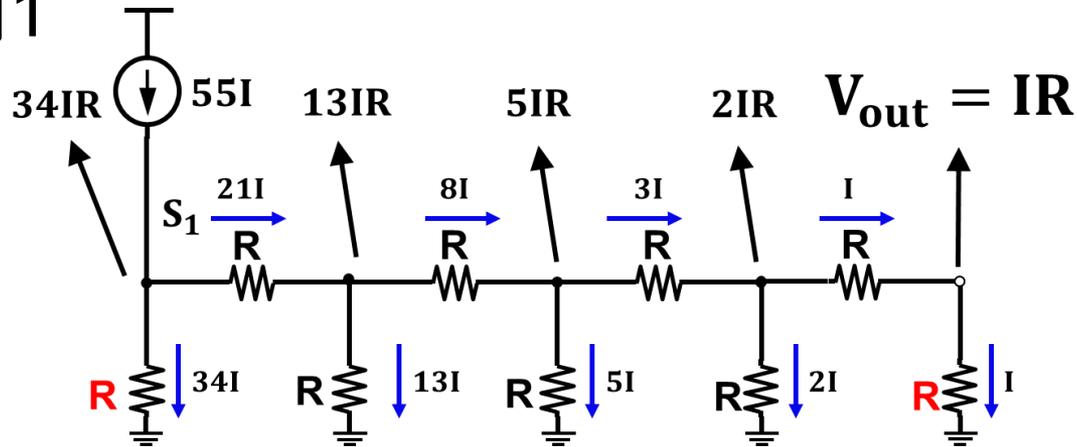
電流を引き抜き
出力ノードの電流減少

加算と減算が1対1で対応⇒従来のSARロジックを使用可能

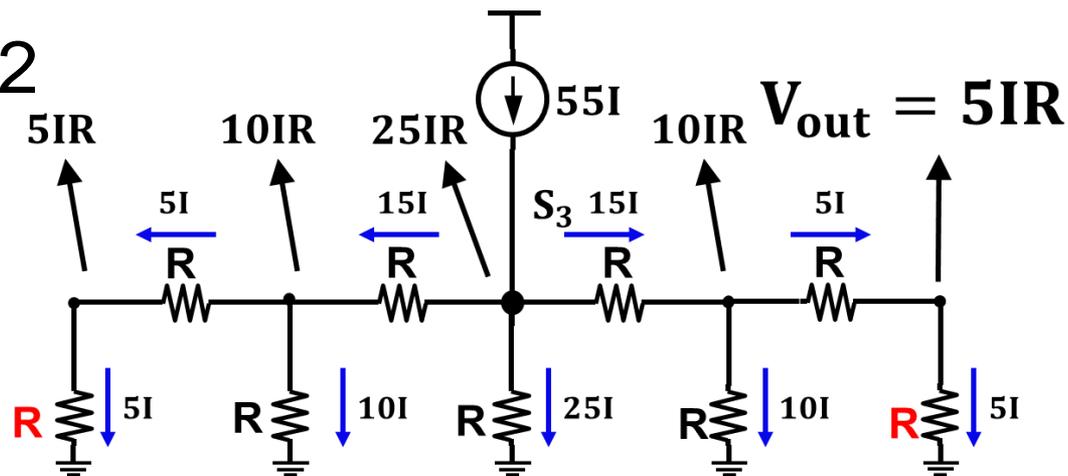


加算動作

加算動作例1



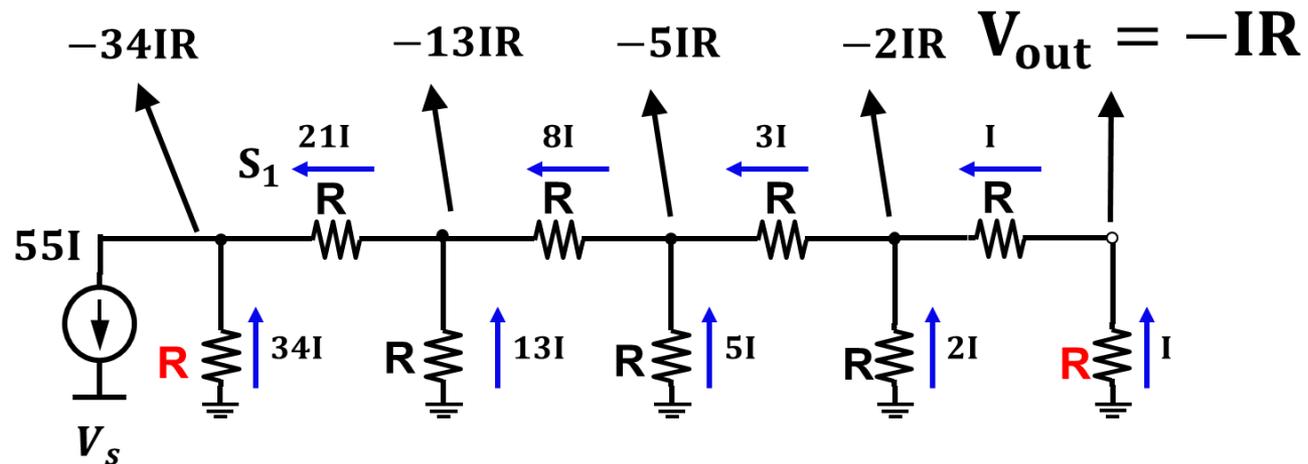
加算動作例2



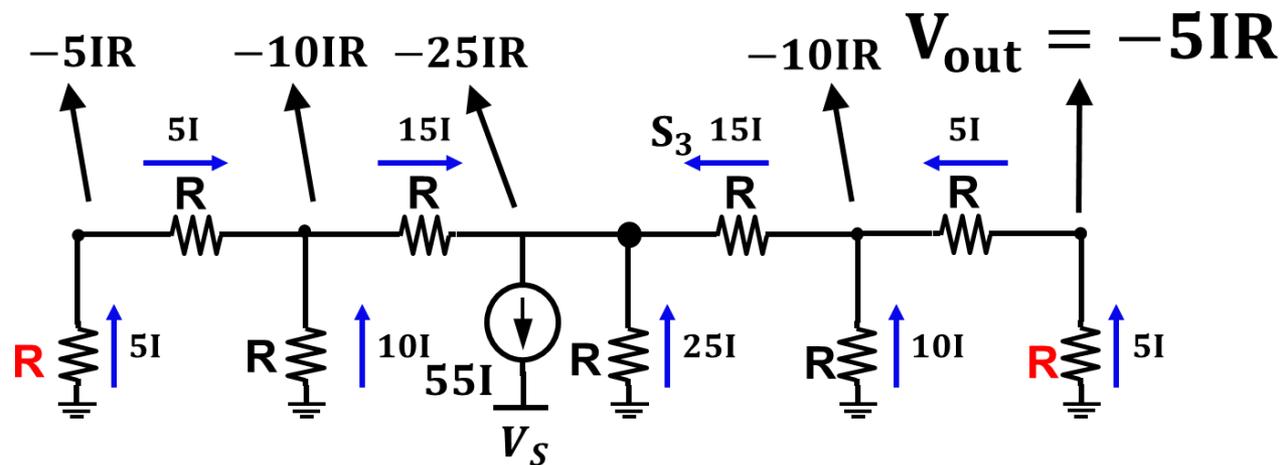
電流源の位置を変えることで加算量を可変

減算動作

減算動作例1



減算動作例2



電流源の位置を変えることで減算量を可変

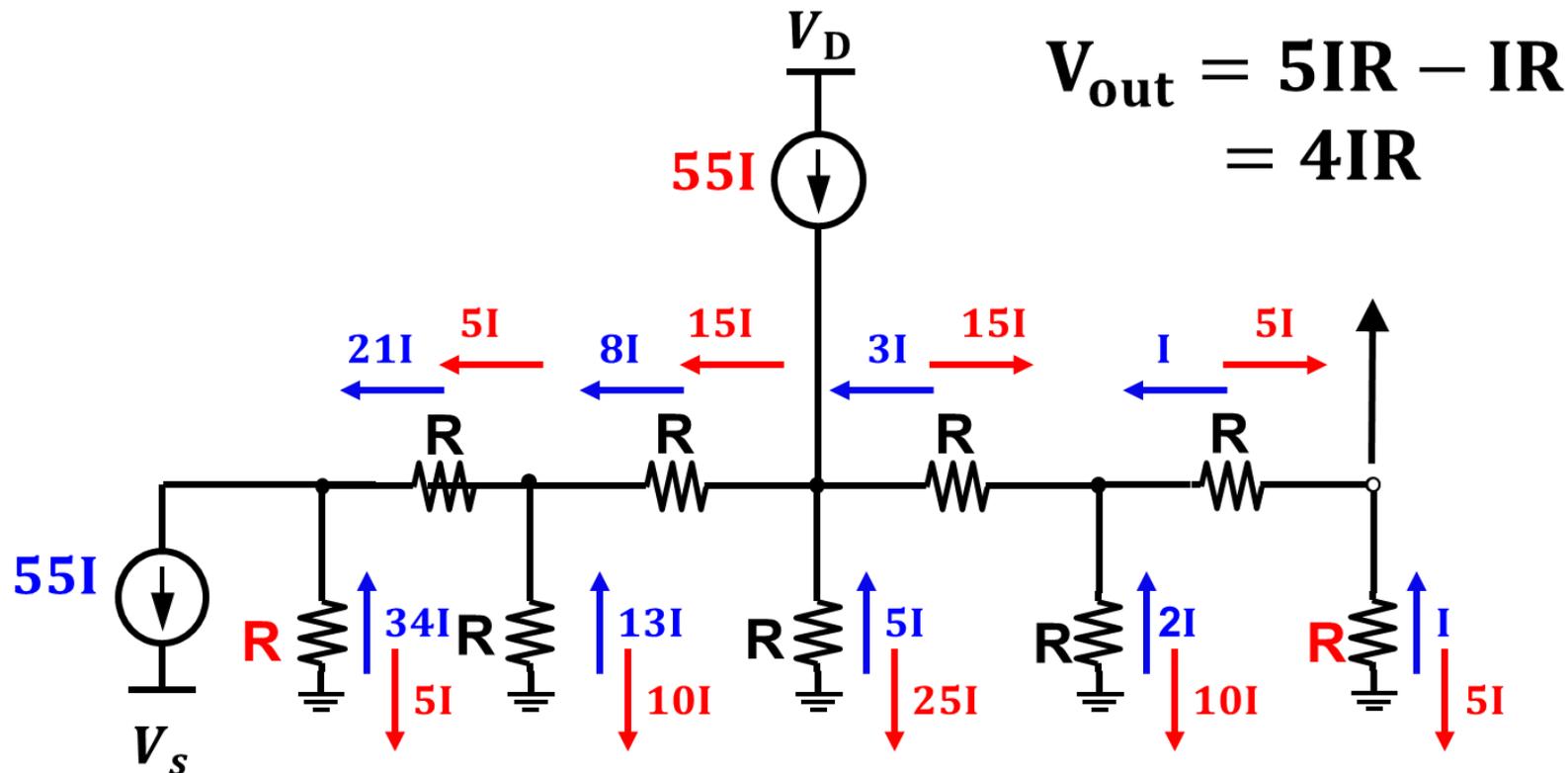
加減算動作

加減算動作例

⇒線形回路のため重ね合わせの理を使用

⇒加算電流と減算電流を分離して検討し、

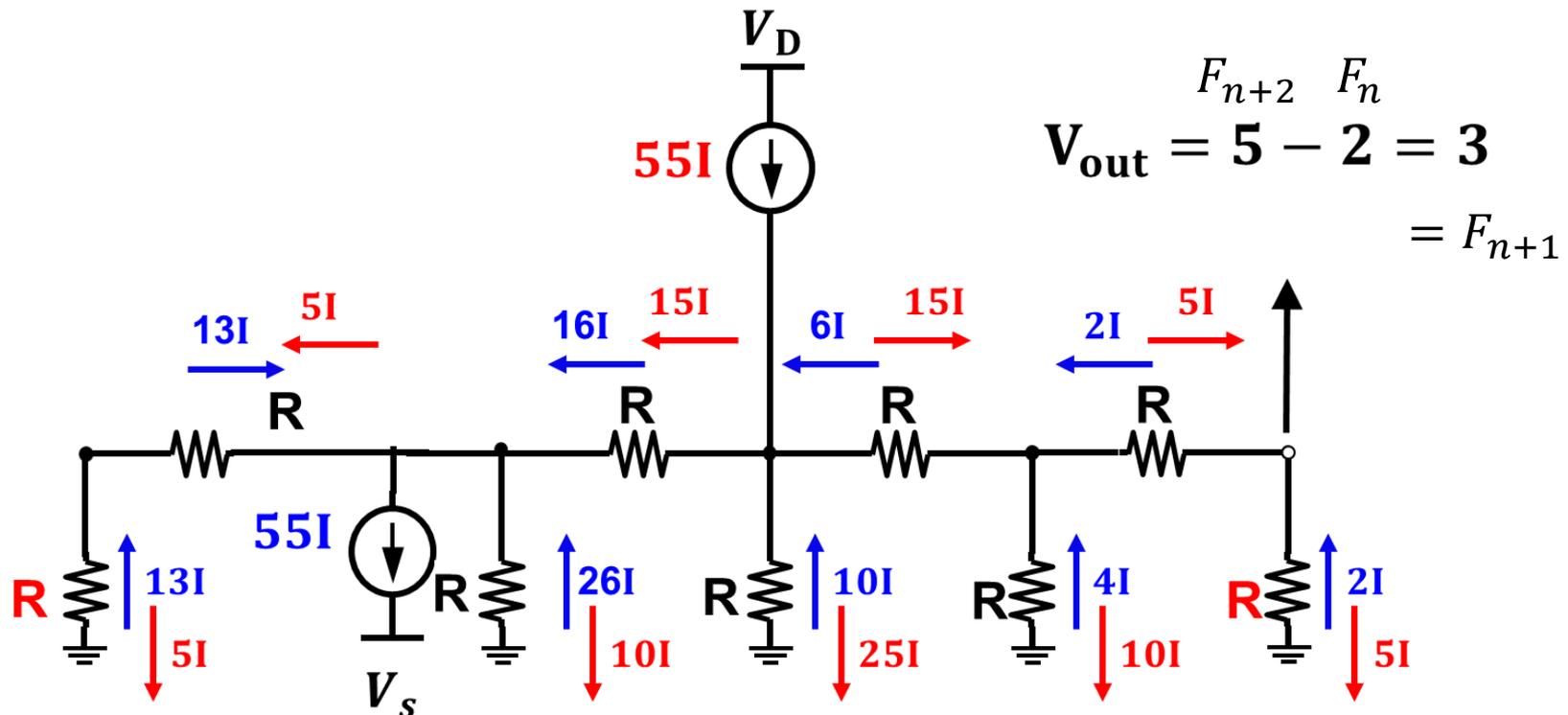
出力ノードで電流による減算が可能に



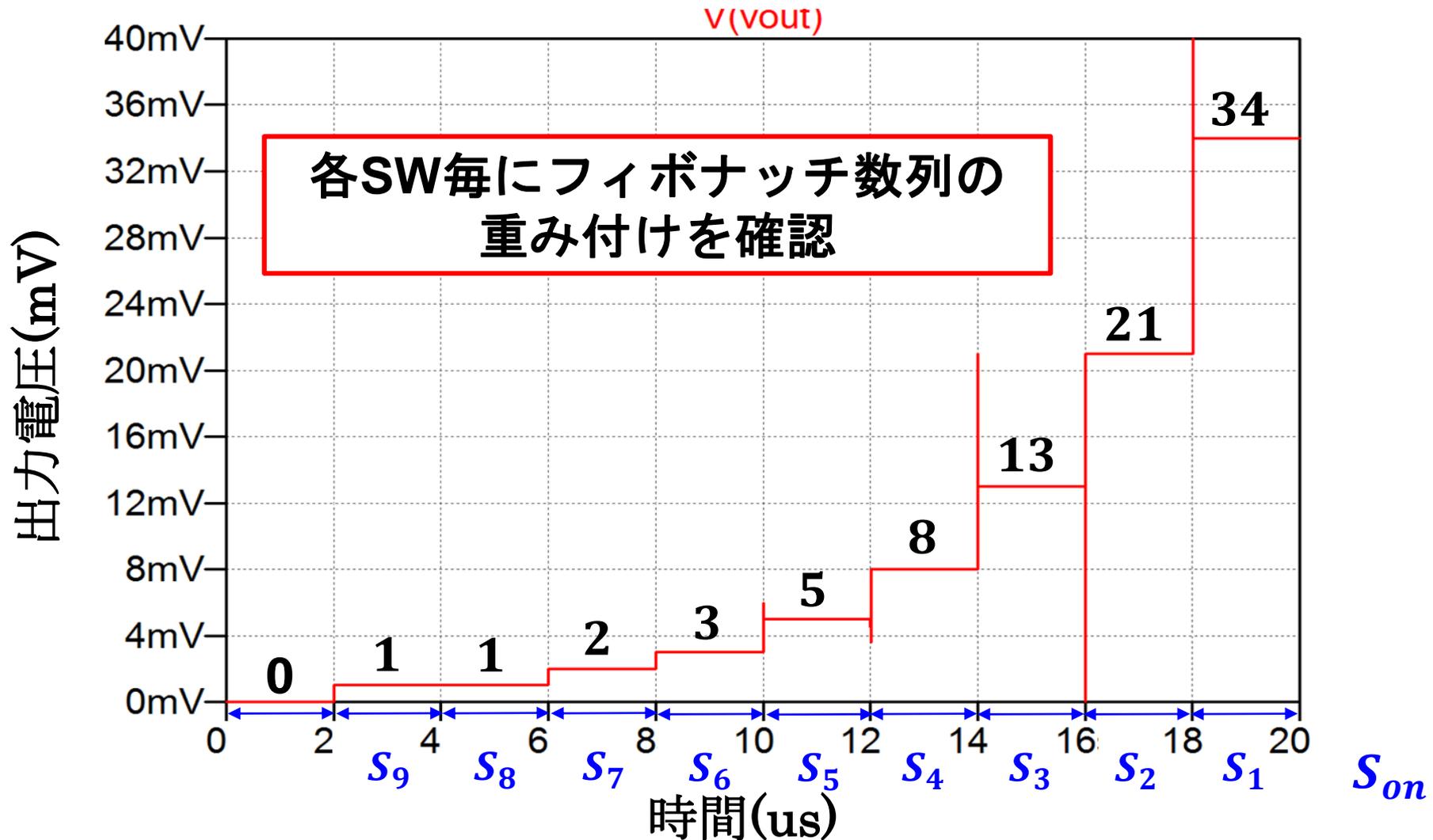
フィボナッチ奇数項

偶数項ラダー回路

- ⇒ 偶数項の減算処理により奇数項の表現が可能に
- ⇒ 奇数項1つで1組(加算/減算)の電流源対を追加するのみ
(ラダーを1列にでき、加算部が不要なことから回路の小規模化に貢献)



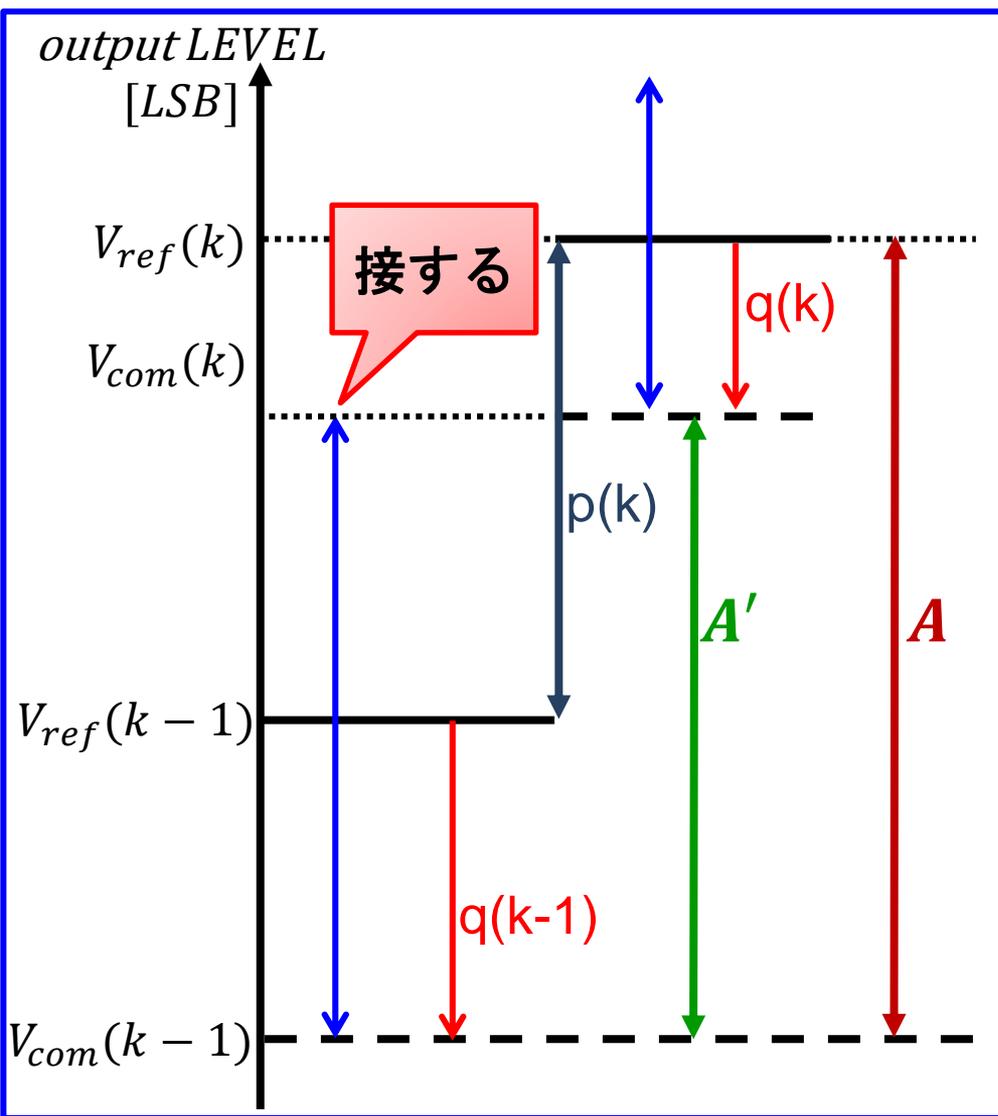
重み付けの確認



各スイッチでフィボナッチ重みが出力

フィボナッチ設計手法の整定時間

フィボナッチDACの整定時間を考える フィボナッチ手法 (1.618進)



Step	1st	2nd	3rd
Weight $p(k)$	16	8	5
	31		
	30		
	29		
	28		
	26		
	25		
	24		
LEVEL	23		
	22		
	21		
	20		
	19		
	18		
	17		
	16		
	15		

V_{DAC}

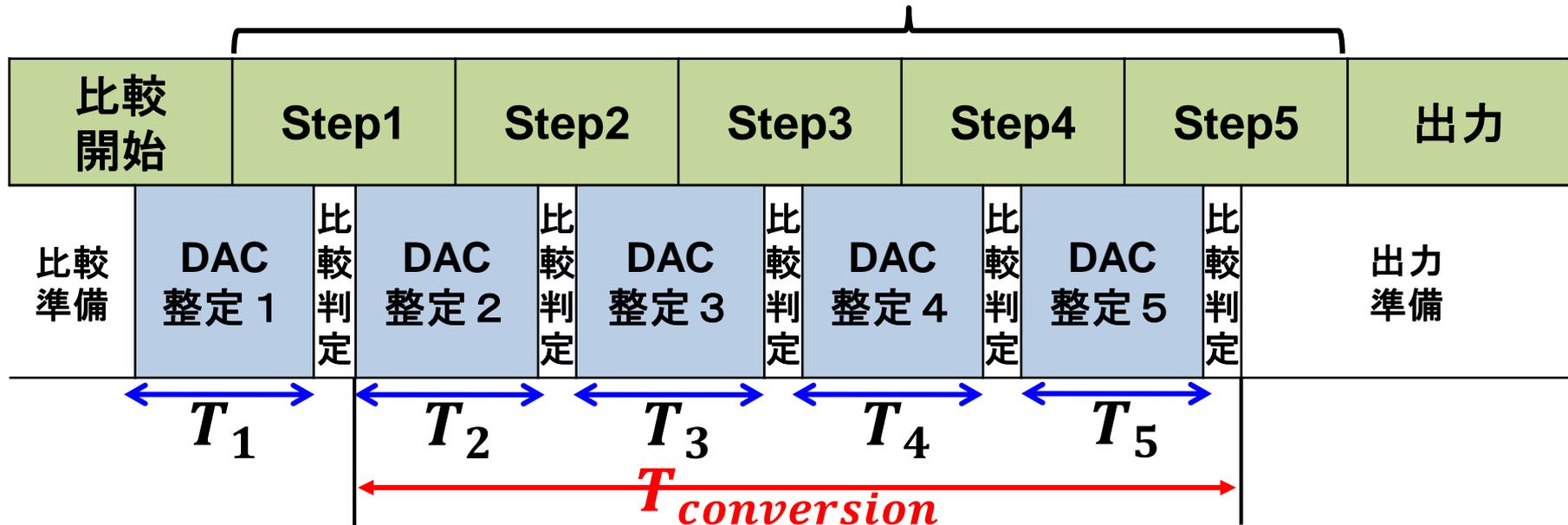
変換時間シミュレーション

5step SAR ADC変換動作

比較判定フェーズ

フェーズ

時間



シミュレーション条件(基数法と整数論法の整定時間)

- 整定時間公式利用・比較判定時間無視

$$T_k = \tau \ln \left(\frac{1}{1-x} \right) = \tau \ln \left(\frac{p(k) + q(k-1)}{q(k)} \right)$$

- $q(k)=0$ [LSB]なら $q(k)=0.5$ [LSB]とみなす
- $T_1=0$:考慮しない(全方式で同じ)、 T_2 :異なる計算式を利用
- 時間の計算結果がマイナスのときは $T=0$ とする

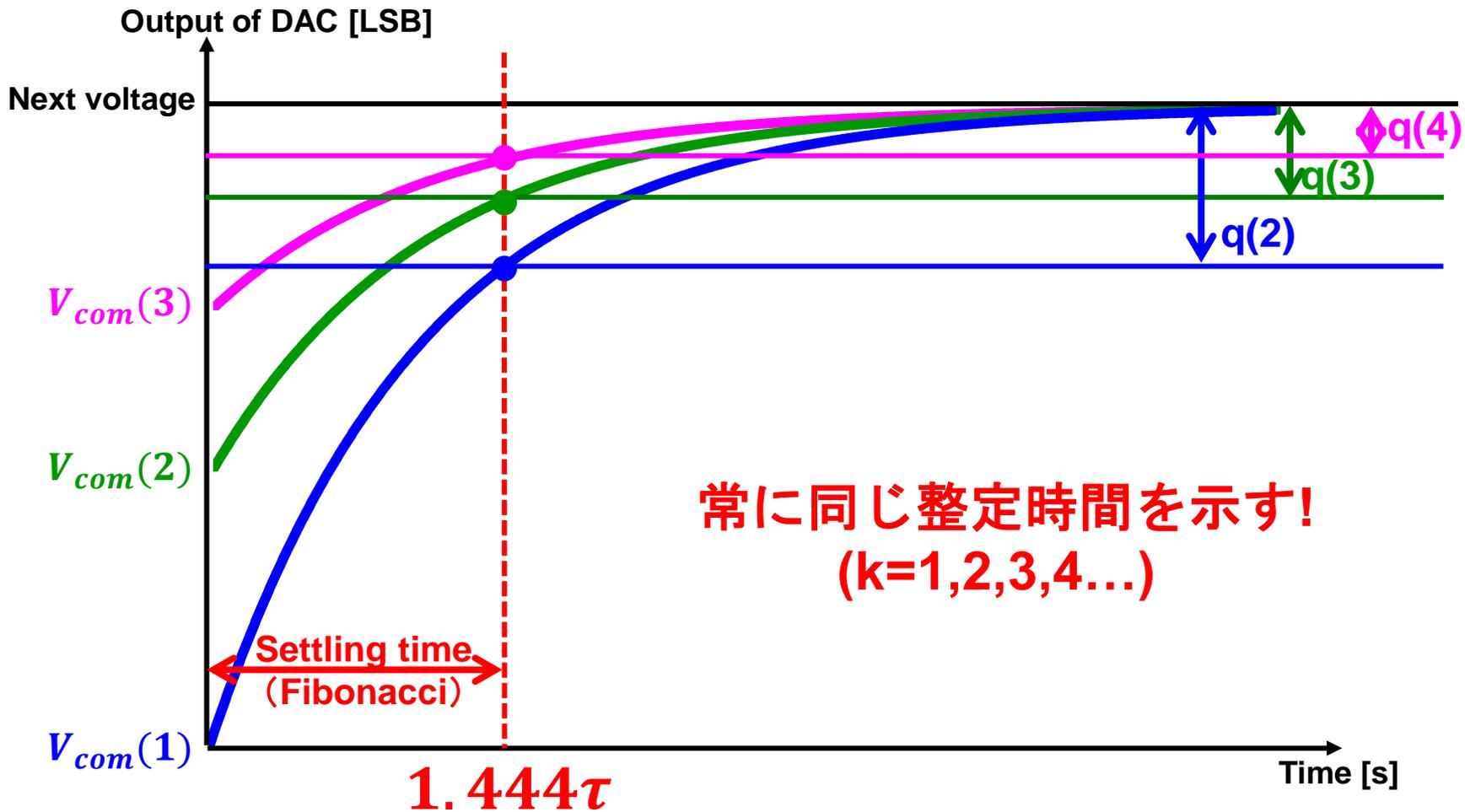
- ・ 可変クロック
 - ・ 固定クロック
- 変換時間調査

フィボナッチ数列の一般整定時間

新たな性質の発見！

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

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



フィボナッチ設計手法の整定時間

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	↑↓	↑↓		↑↓	↑↓		

フィボナッチ数列SAR ADC性質

①許容値 $q(k)$ は必ずフィボナッチ数

$$q(k) = F_{M-k-1}, q(k-1) = F_{M-k}$$

②許容できる範囲が必ず接する

x は必ず $q(k), q(k-1)$ でのみ決定

整定時間 T は

$$T = \tau \ln(2\varphi + 1)$$

$$= 1.444\tau$$

フィボナッチ数列では
可変クロックを用いたとしても
(LSB側を除いて)常に一定!

※1. リュカ数も同様に一定時間実現

※2. 一般手法では $p(k)$ と $q(k)$ に
関連性がないためできない

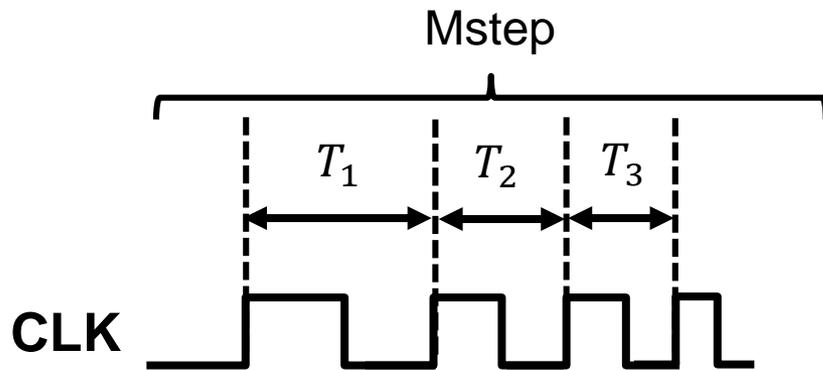
DAC出力整定の一般化

整定時間

$$T = \tau \ln\left(\frac{1}{1-x}\right) = \tau \ln\left(\frac{p(k) + q(k-1)}{q(k)}\right)$$

可変クロックAD変換

T の単純な合計値

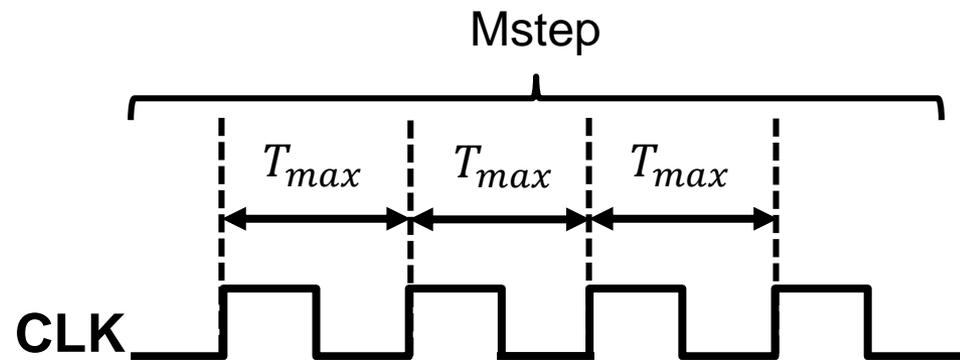


AD変換時間(可変)

$$T_{conv_vari} = \sum_{i=1}^M T_i$$

固定クロックAD変換

T の最大値 T_{max} のステップ倍
($\Rightarrow x$ が最大のところ)

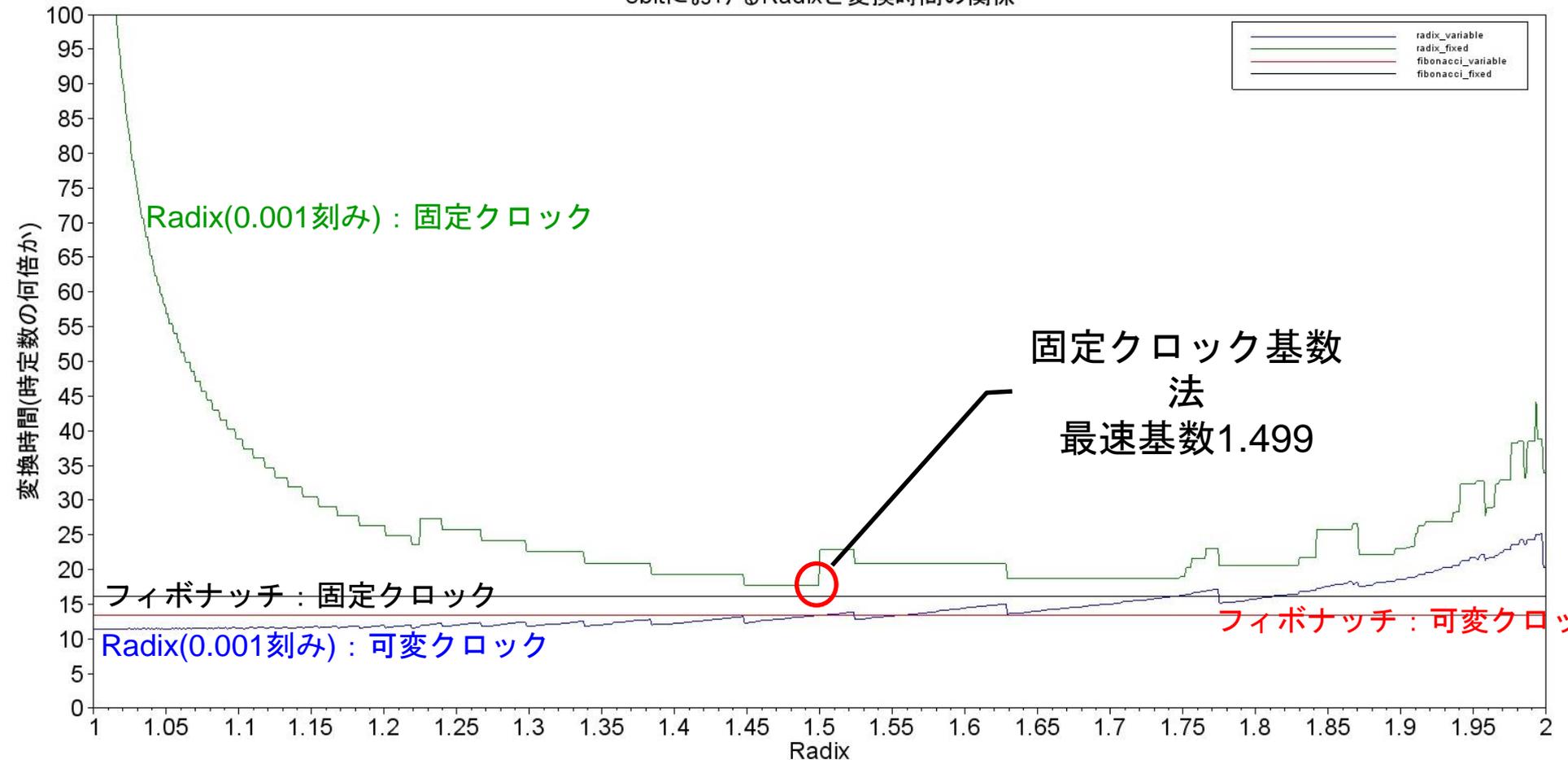


AD変換時間(固定)

$$T_{conv_fixed} = T_{max} \times M$$

8bitフィボナッチ数シミュレーション結果

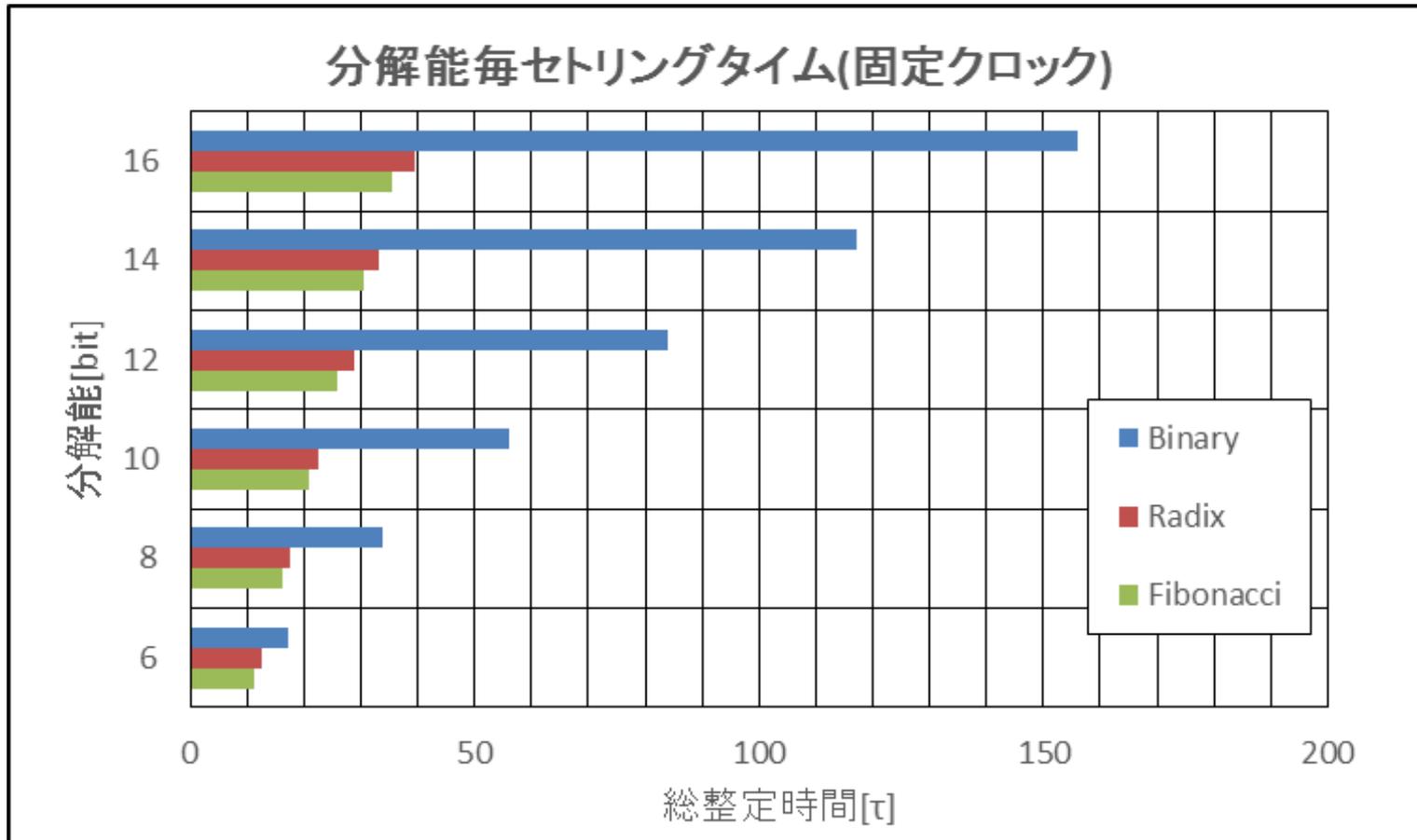
8bitにおけるRadixと変換時間の関係



固定クロックにおいてはフィボナッチ数列利用が最速！

※ただし証明ができていないため現時点での最速

不完全整定時間比較



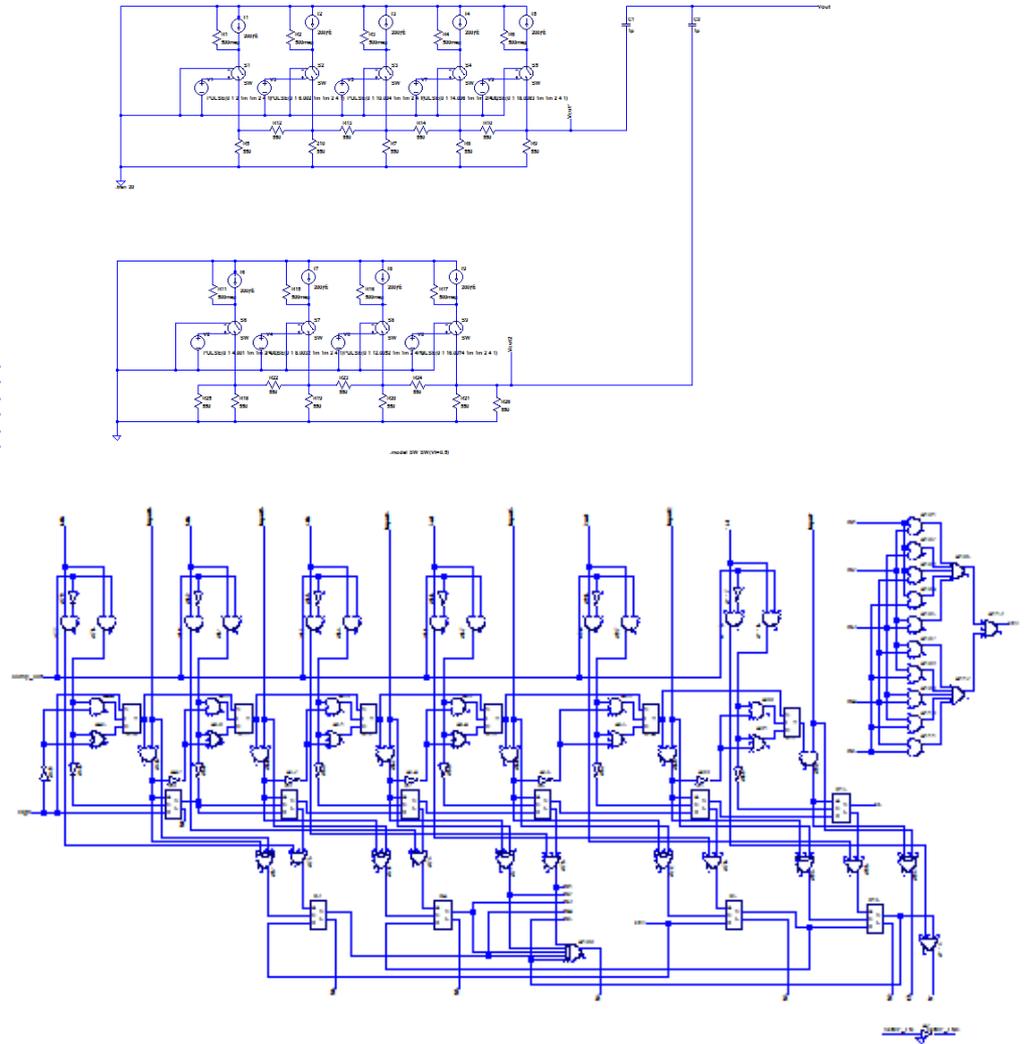
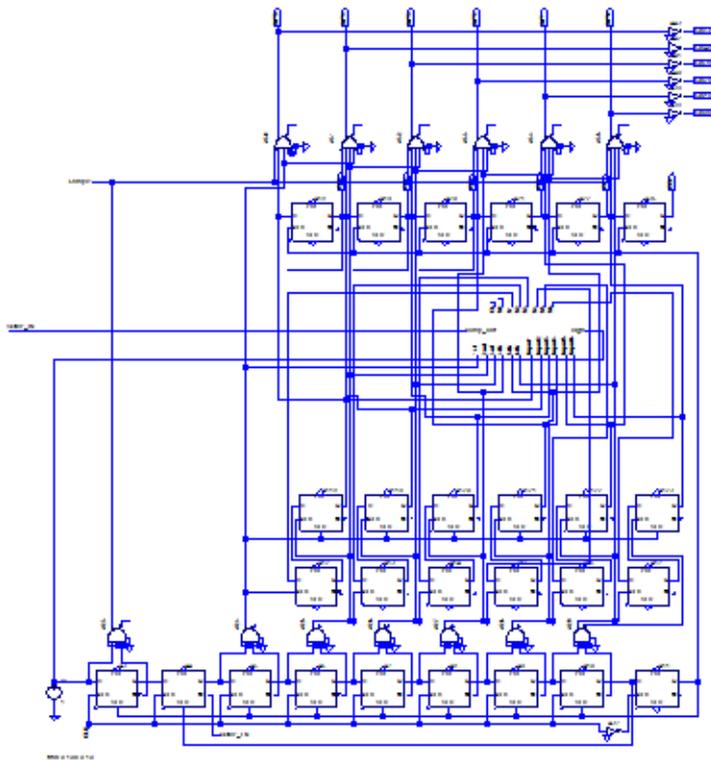
Radix手法から

約1~3 τ 、最大で**4 τ 以上**短縮

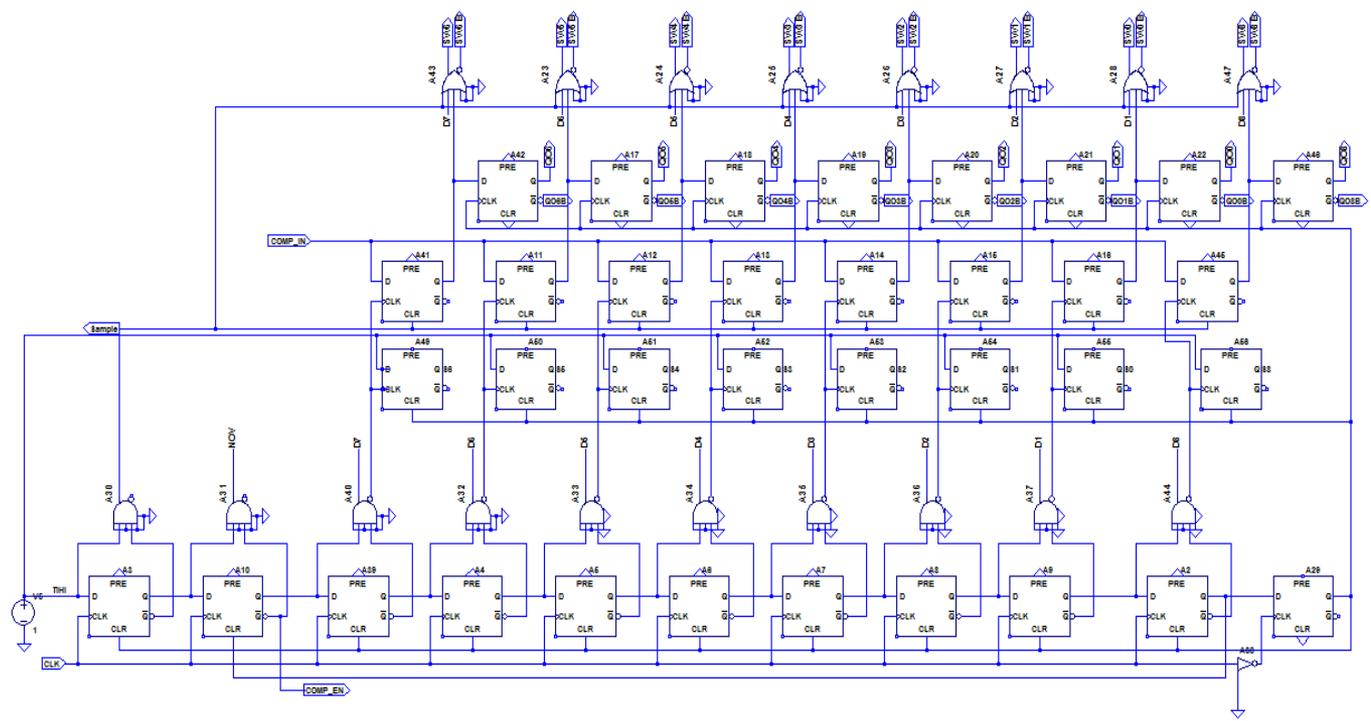
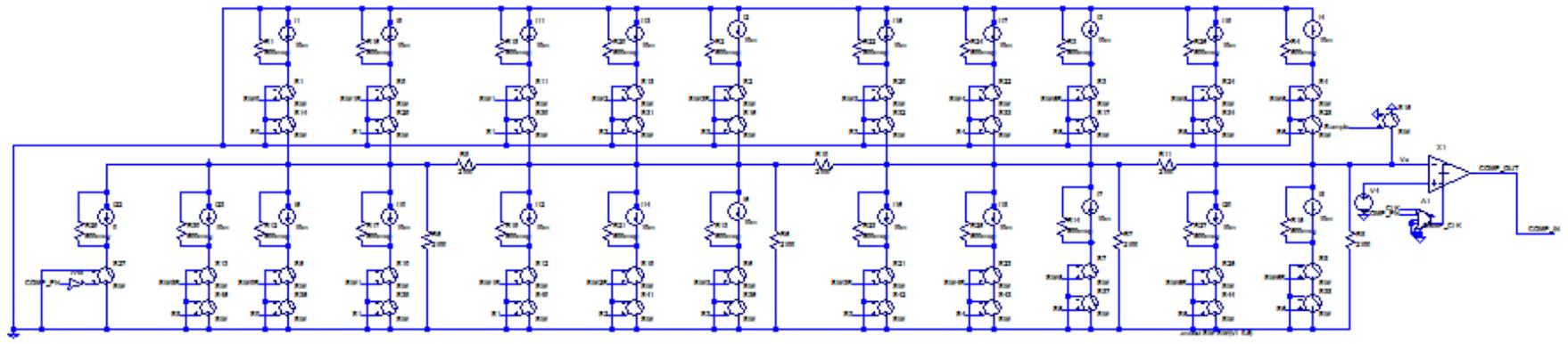
最大で非冗長の1/5の時間で整定

**全分解能固定クロックで
フィボナッチ手法は最も高
速！**

従来のSARADC



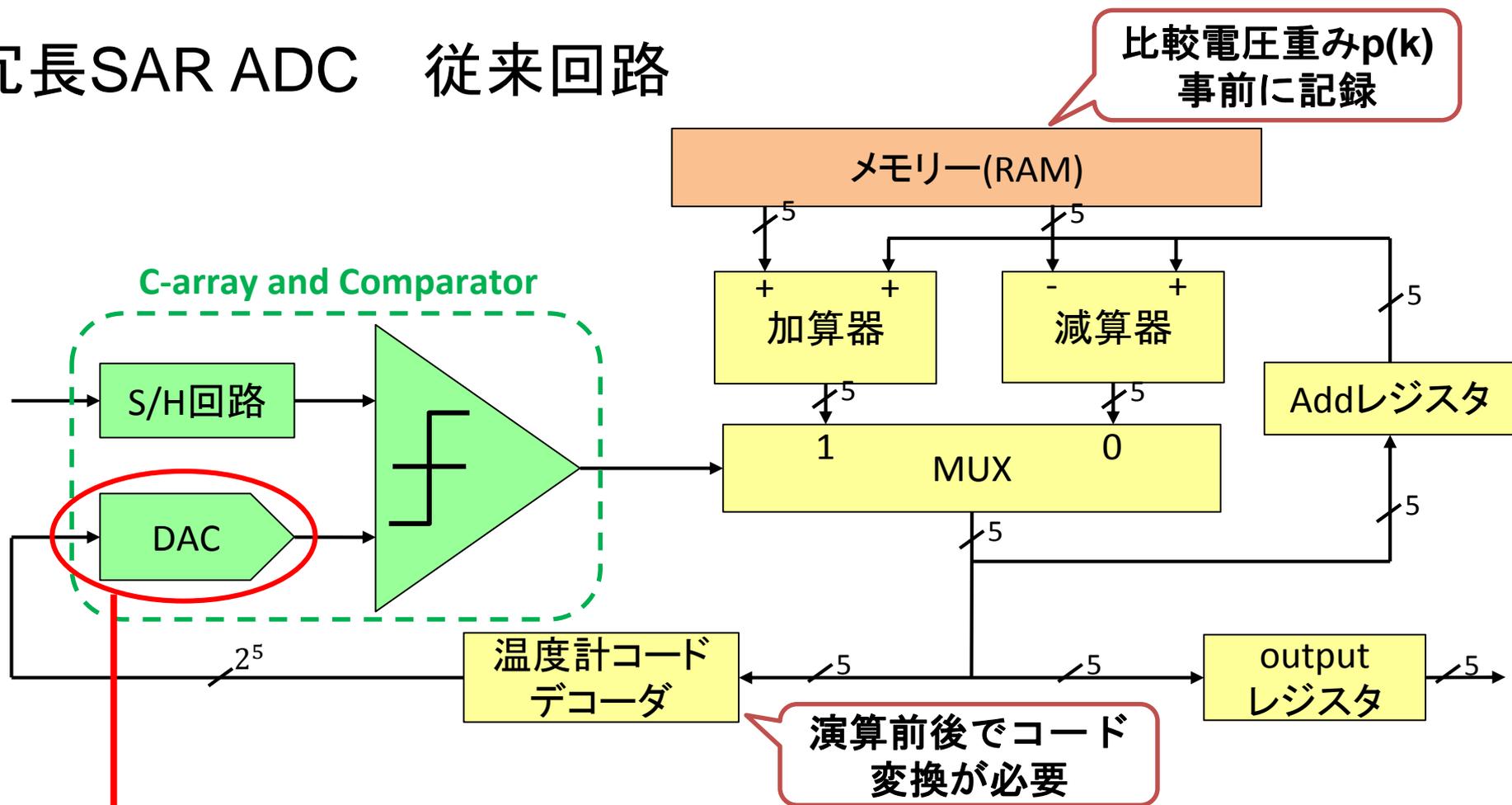
提案SARADC



冗長SAR ADCの実現

手法2 ⇒ 演算前後のコード変換が必要なため遅延や回路規模の増大

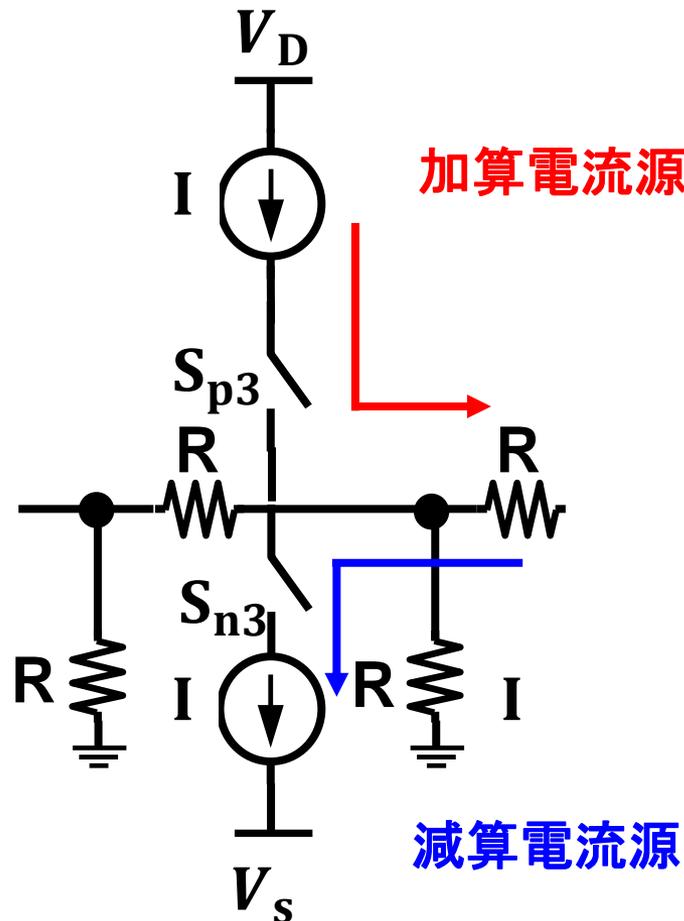
冗長SAR ADC 従来回路



フィボナッチDACを作成し、回路を簡略化・小型化できないか？

2つの逐次探索方法

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



加減算型フィボナッチDAC

値を加算するか減算するかを判定

- ・ 二進SARロジックを使用可能
- ・ 抵抗ラダー列を1段に
- ・ 加算部が不要

