

集積回路システム工学  
講義資料

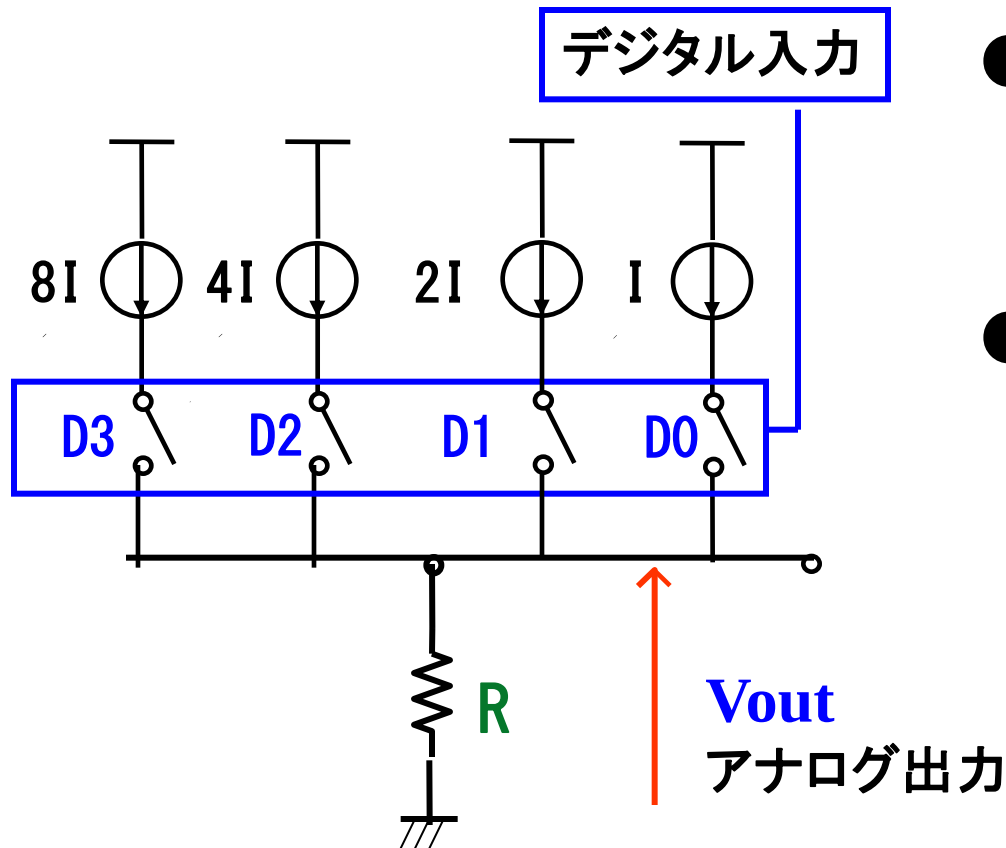
# AD/DA 変換器の失敗事例

教科書にでているのはうまくいった技術のみ

群馬大学 小林春夫



# 電流型2進重み付けDA変換回路 (回路)



## ● メリット

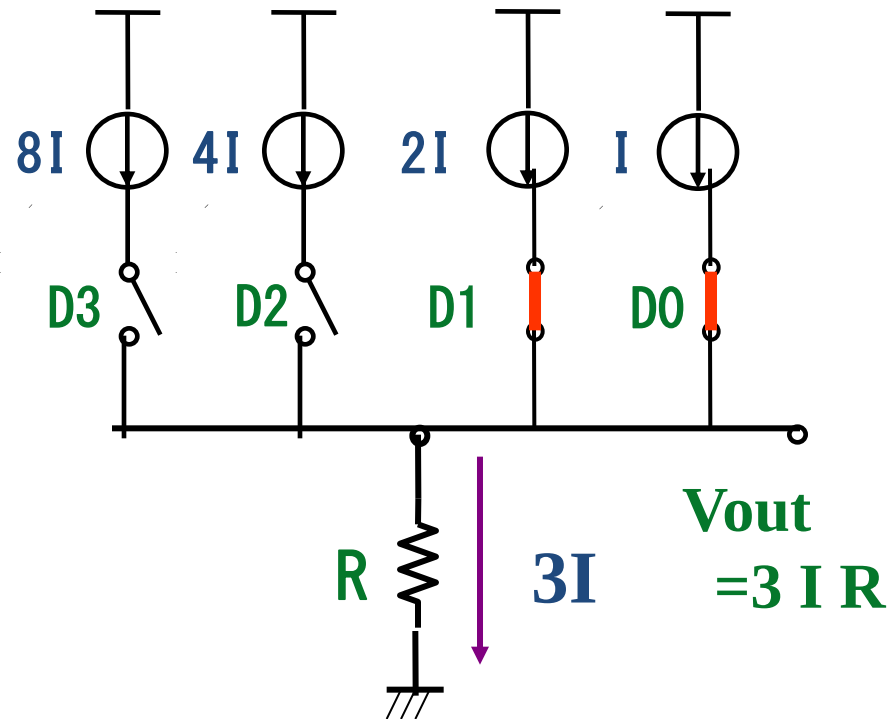
- ・回路規模が小さい
- ・サンプリング速度が速い

## ● デメリット

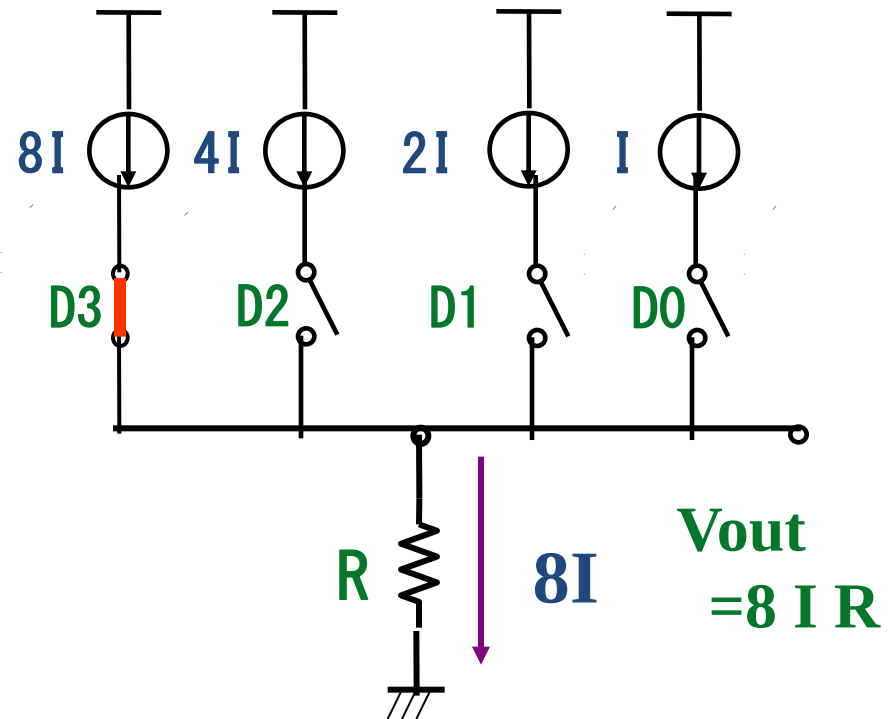
- ・グリッチが大きい
- ・入出力間の単調性が  
確保出来ない

# 電流型2進重み付けDA変換回路 (動作)

例: 入力データが3のとき



例: 入力データが8のとき



# 電流型2進重み付けDA変換回路 (原理)



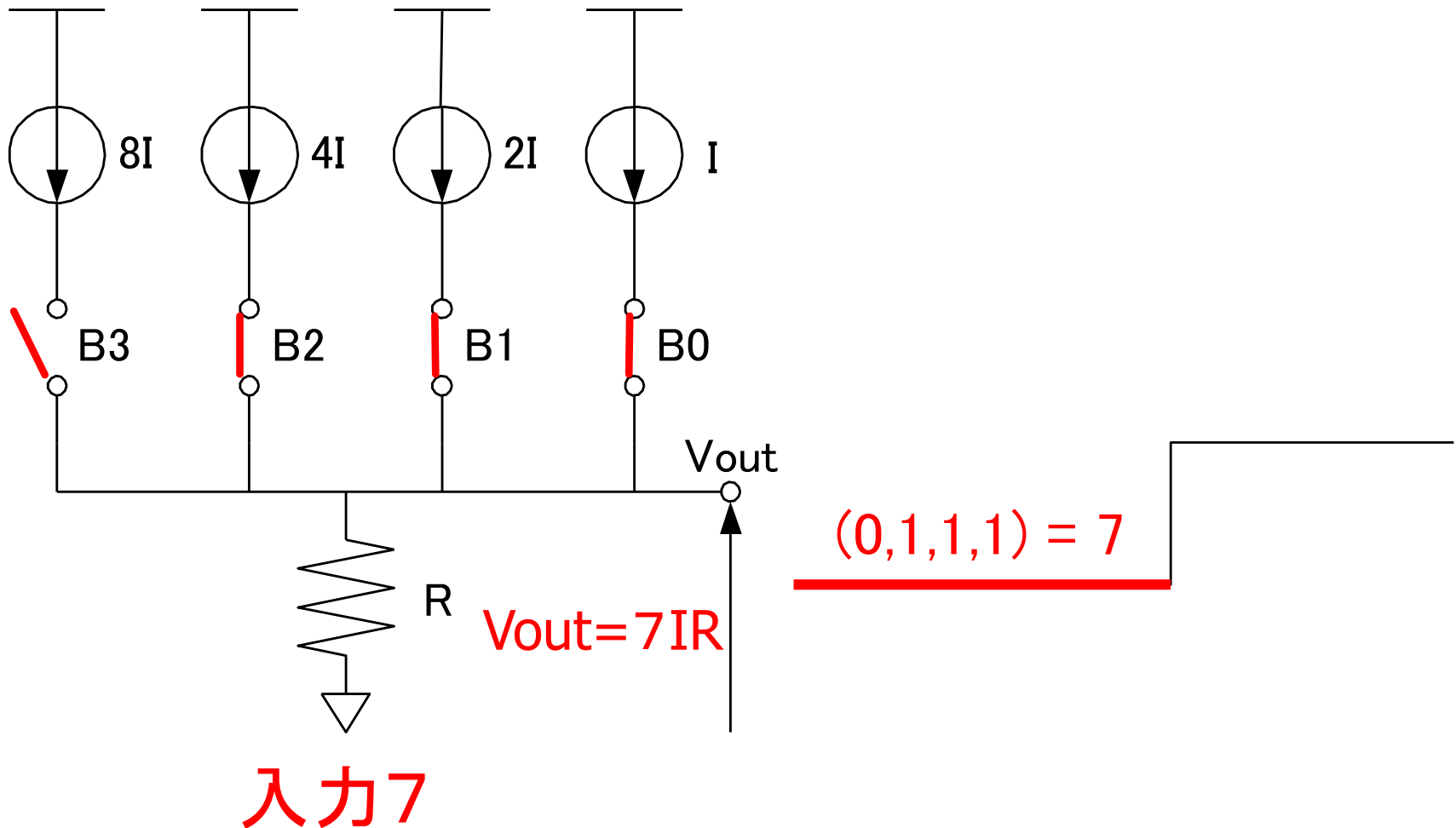
| デジタル<br>入力データ | スイッチ |    |    |    | 出力   |
|---------------|------|----|----|----|------|
|               | D3   | D2 | D1 | D0 | Vout |
| 0             | 0    | 0  | 0  | 0  | 0    |
| 1             | 0    | 0  | 0  | 1  | 1R   |
| 2             | 0    | 0  | 1  | 0  | 2R   |
| 3             | 0    | 0  | 1  | 1  | 3R   |
| 4             | 0    | 1  | 0  | 0  | 4R   |
| 5             | 0    | 1  | 0  | 1  | 5R   |
| 6             | 0    | 1  | 1  | 0  | 6R   |
| 7             | 0    | 1  | 1  | 1  | 7R   |
| 8             | 1    | 0  | 0  | 0  | 8R   |
| ⋮             |      |    | ⋮  |    | ⋮    |
| 15            | 1    | 1  | 1  | 1  | 15R  |

スイッチ 1のとき ON  
0のとき OFF

デジタル入力データに  
比例したアナログ出力  
Vout が生成される。

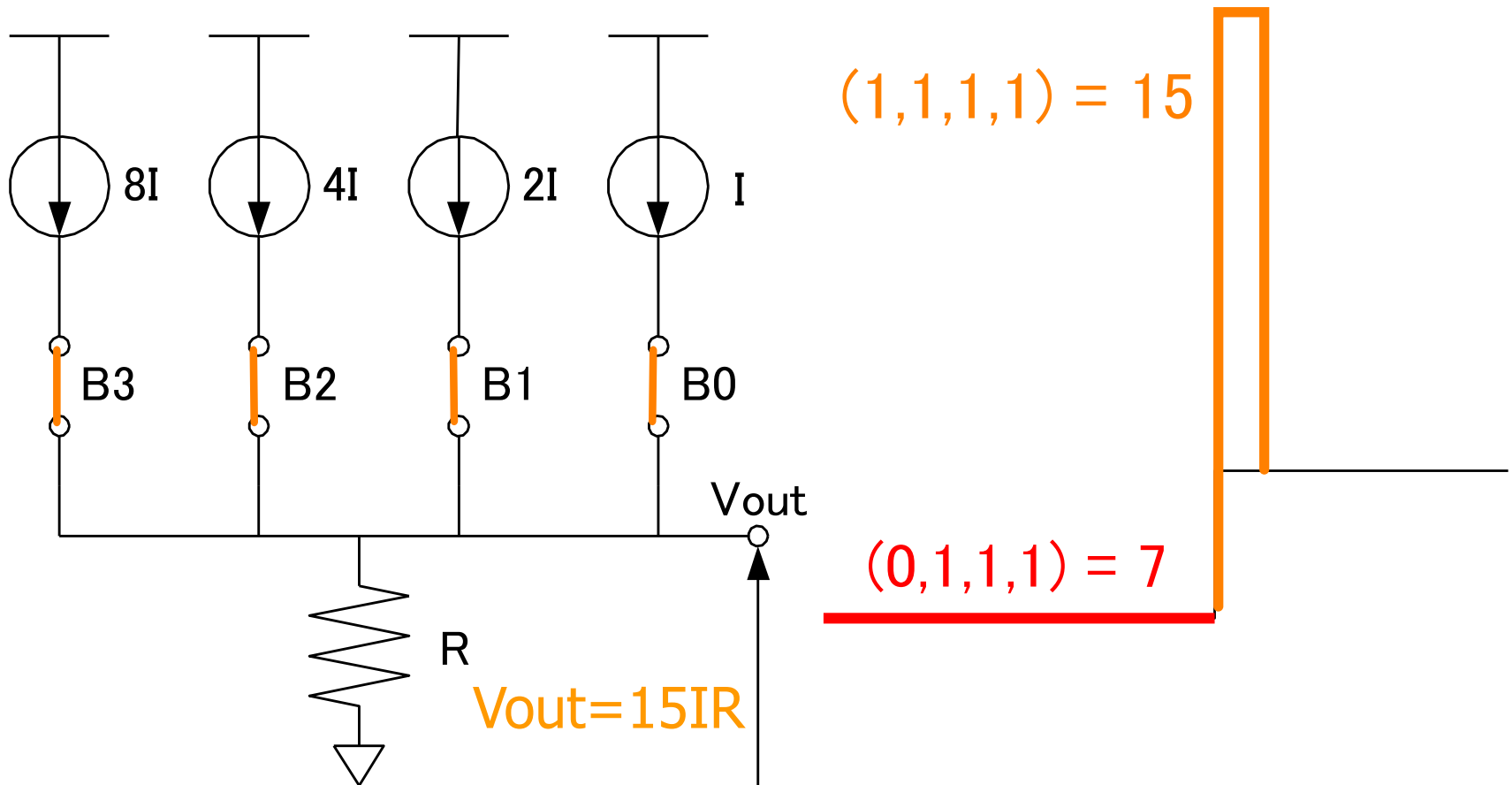
# スイッチ切り替えタイミング

スキューが有る場合



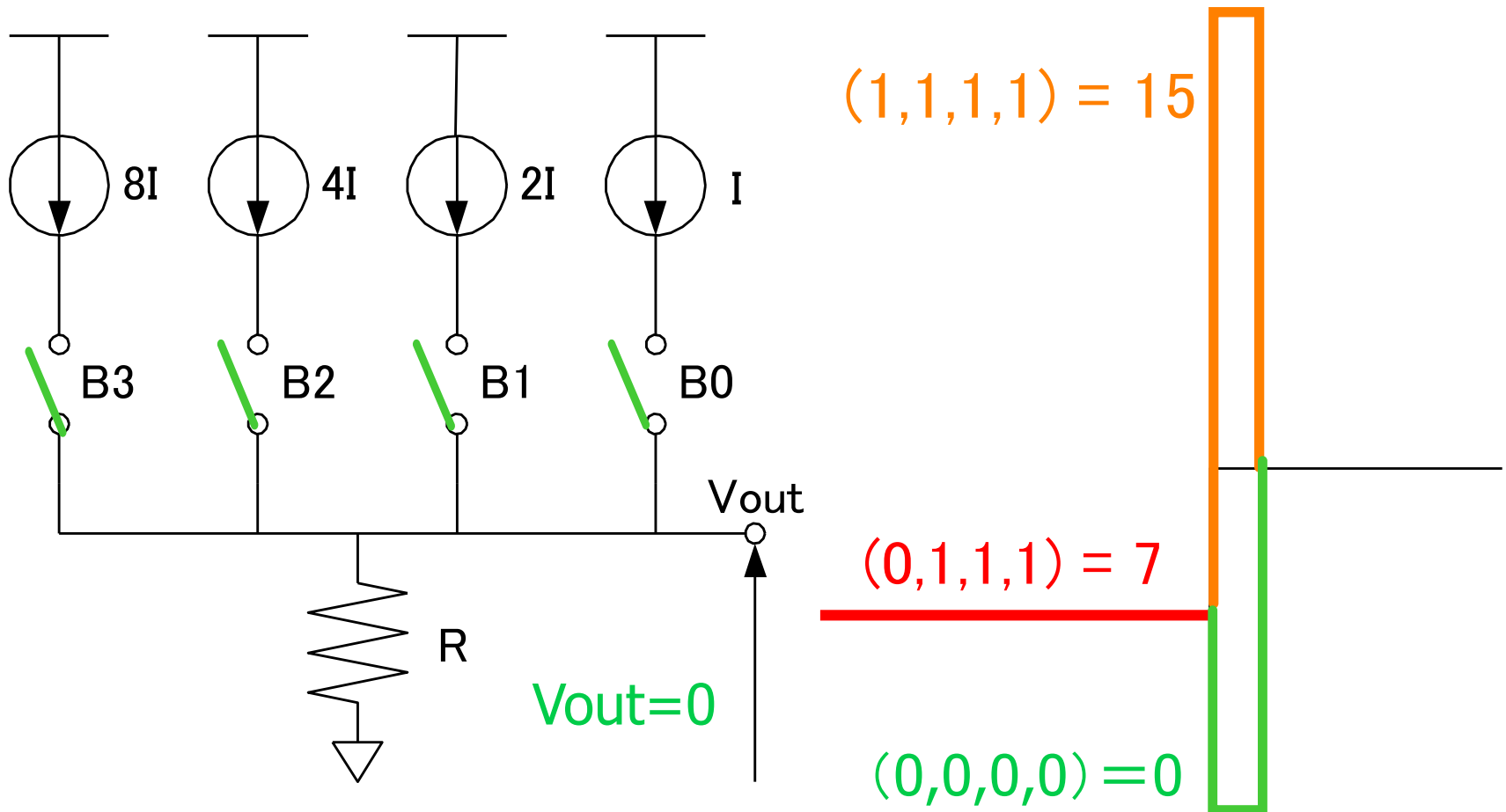
# スイッチ切り替えタイミング

スキューが有る場合



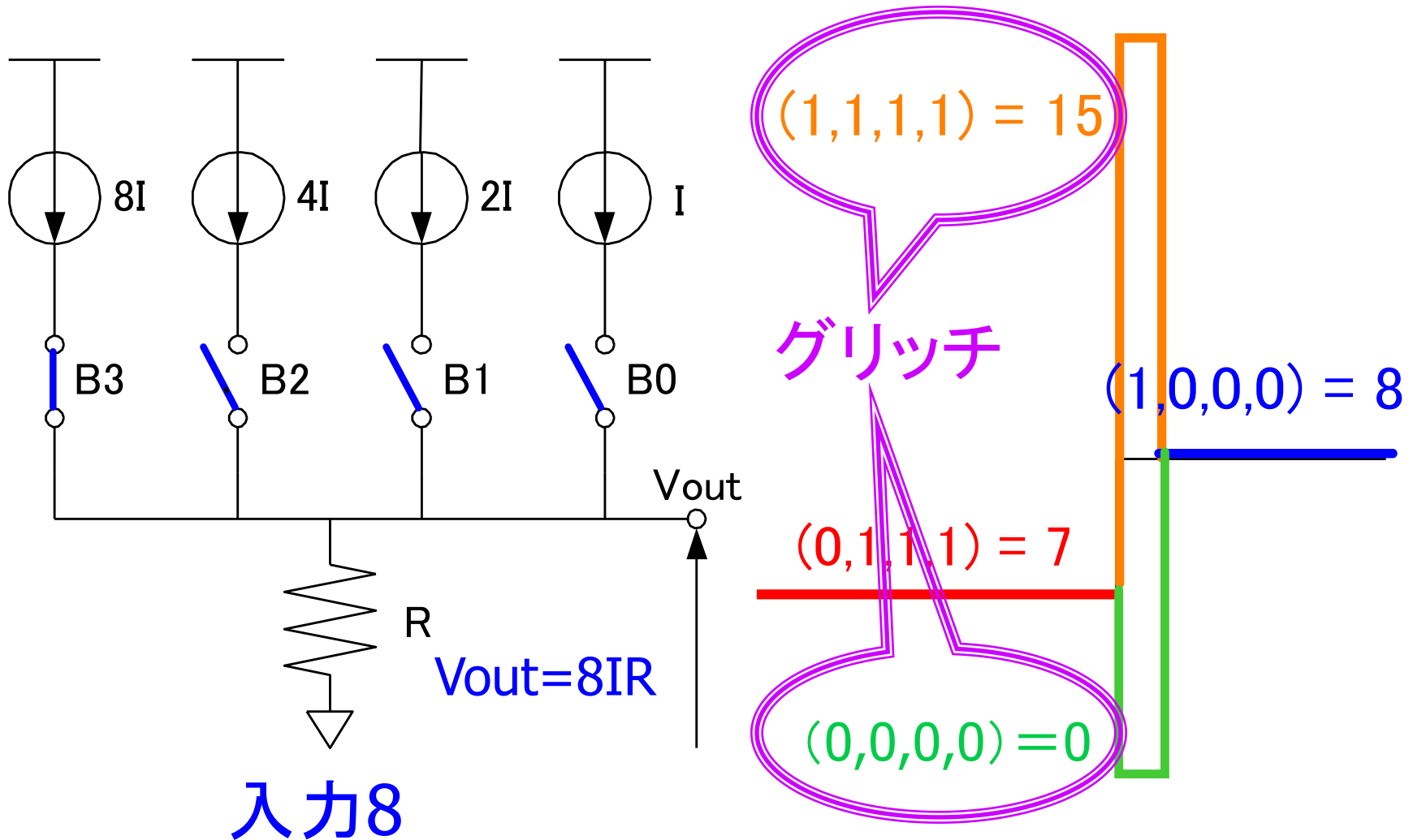
# スイッチ切り替えタイミング

スキューが有る場合



# スイッチ切り替えタイミング

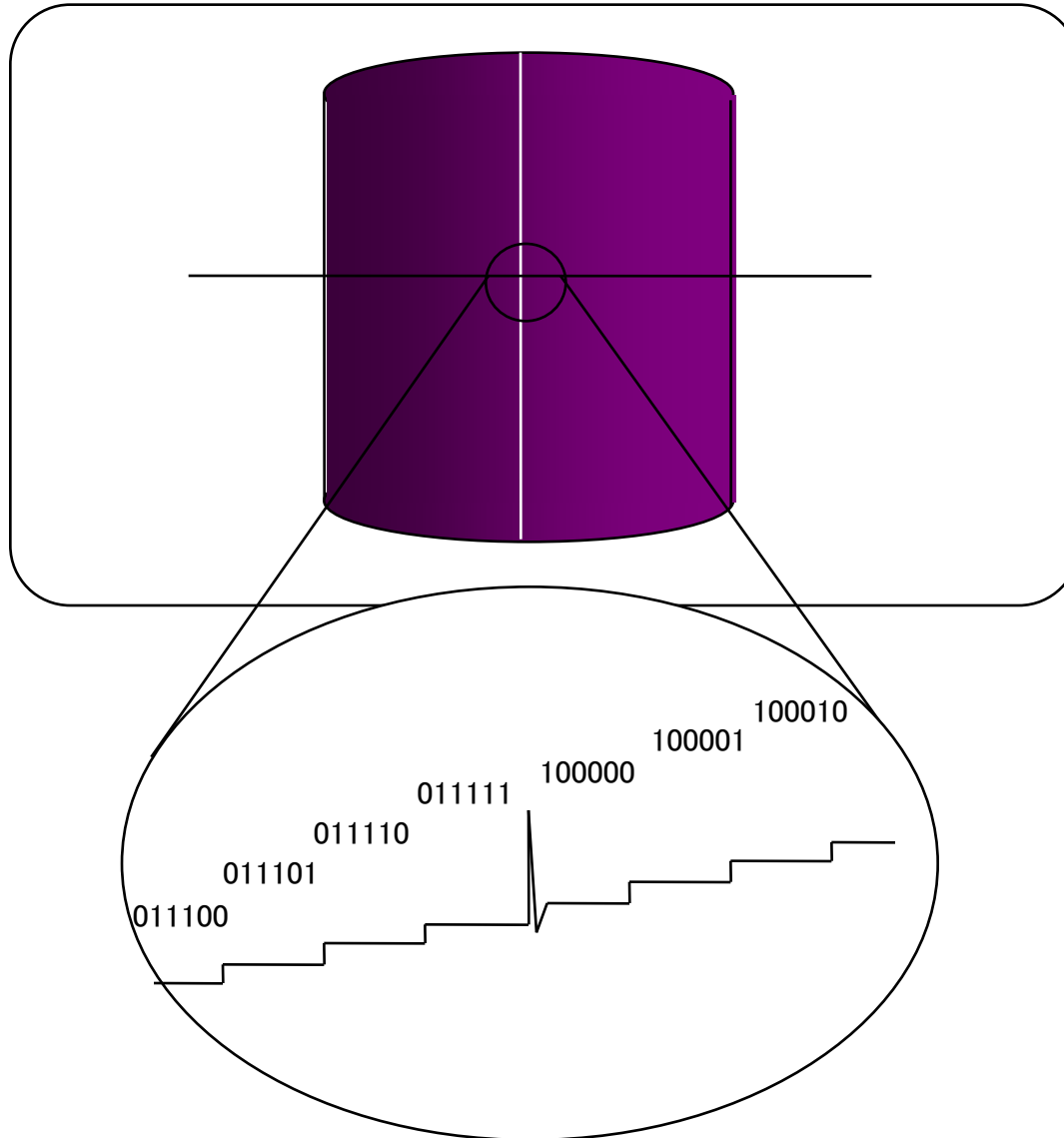
スキューが有る場合





# DA変換器におけるグリッチの影響

## グラフィックディスプレイ用



人間の視覚は  
グリッチに敏感

# Binary code と Gray code

| Decimal numbers | Binary Code | Gray Code |
|-----------------|-------------|-----------|
| 0               | 0000        | 0000      |
| 1               | 0001        | 0001      |
| 2               | 0010        | 0011      |
| 3               | 0011        | 0010      |
| 4               | 0100        | 0110      |
| 5               | 0101        | 0111      |
| 6               | 0110        | 0101      |
| 7               | 0111        | 0100      |
| 8               | 1000        | 1100      |
| 9               | 1001        | 1101      |
| 10              | 1010        | 1111      |
| 11              | 1011        | 1110      |
| 12              | 1100        | 1010      |
| 13              | 1101        | 1011      |
| 14              | 1110        | 1001      |
| 15              | 1111        | 1000      |

Gray code の特長

対応する  
decimal numberが  
プラス1  
または  
マイナス1



1つのビットのみ反転

# Gray code とAD変換器

---

Gray code は ロバスト(頑健)



AD変換器は 信号を  
アナログからデジタルの世界へ  
(不連続な領域間の遷移)



AD変換器では  
まず Gray codeに変換し  
その後 Binary code に変換する構成も多い

# Gray code 構成のDA変換器は？

---

AD/DA変換器の研究をはじめたときに

Gray code 構成のDA変換器が実現できれば



グリッチがなくなる。

回路の教科書/論文のどこにも

Gray code 構成のDA変換器は記述されていない。



革新的構成か？

# 専門家に聞く

Asad Abidi 先生 (UCLA)

「Gray code 構成でDA変換器が実現できないか  
多くの研究者が考えたが、  
できない ということが(経験的に)わかった。」



教科書にはこのことは書いていない。  
学会等でも誰も言わない。  
このことから多くのことを学ぶ。

**経験論:** 観察や実験などを繰り返し行うことによって、  
経験を少しずつ積み、結果的に真理に到達する  
フランス・ベーコン

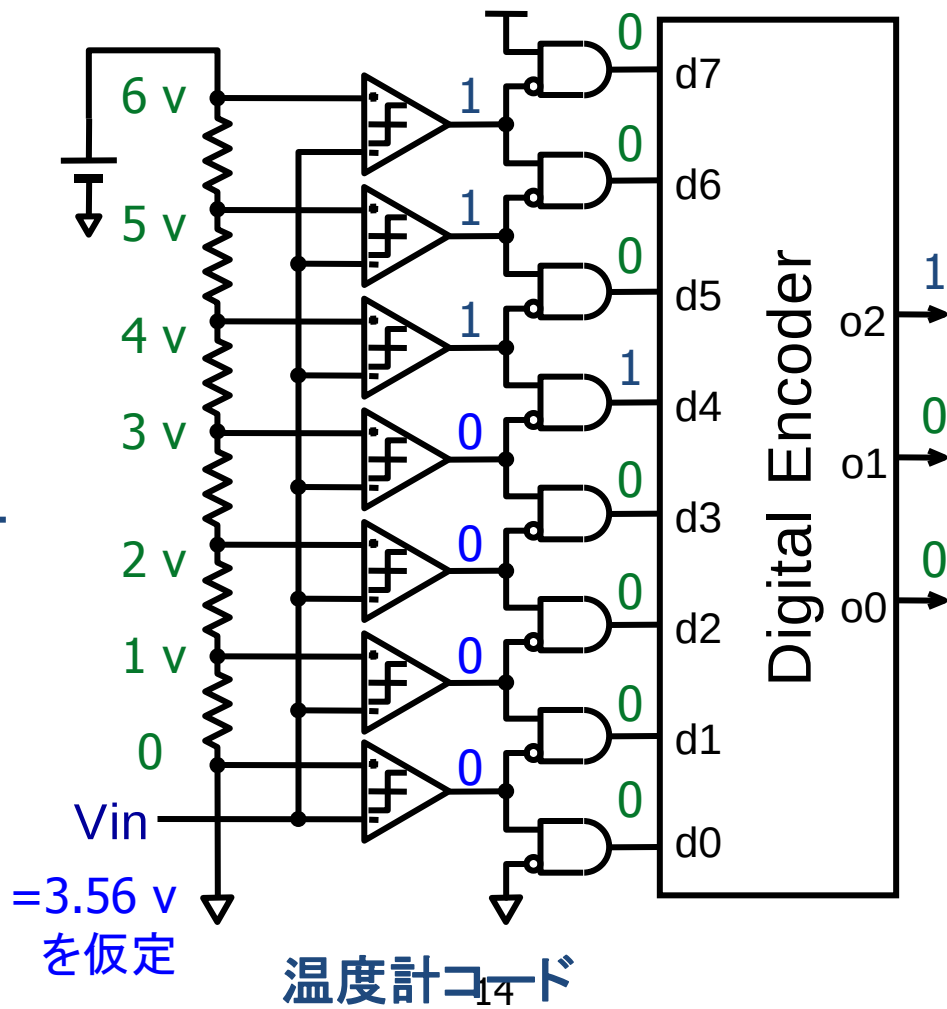
# フラッシュ型ADC

長所 : 高速

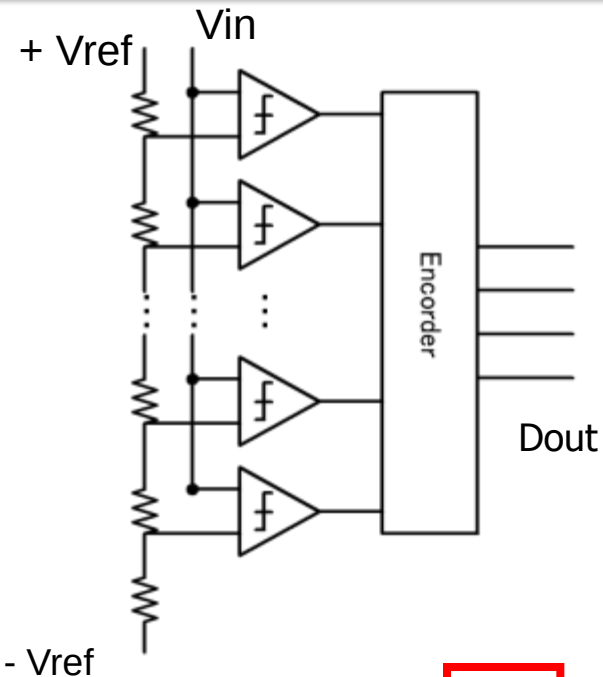
短所 : 回路量 大  
消費電力 大  
入力容量 大

Encoder 真理値表

| d7 | d6 | d5 | d4 | d3 | d2 | d1 | d0 | o2 | o1 | o0 |
|----|----|----|----|----|----|----|----|----|----|----|
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  |
| 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 1  |
| 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 1  | 0  |
| 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 1  | 1  |
| 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 1  | 0  | 0  |
| 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  |
| 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 0  |
| 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  |

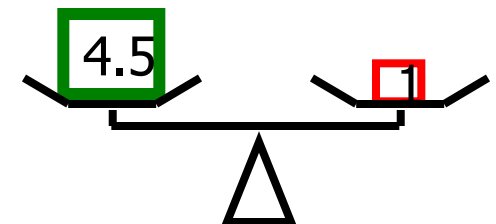
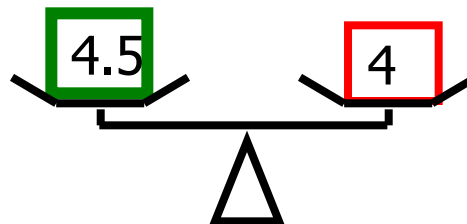
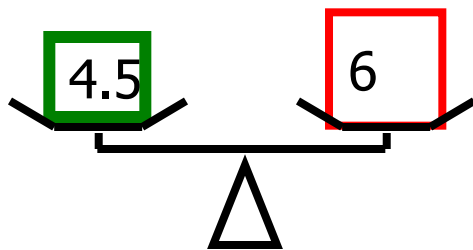
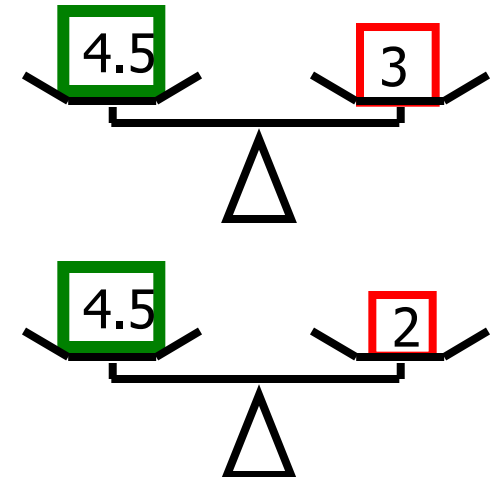
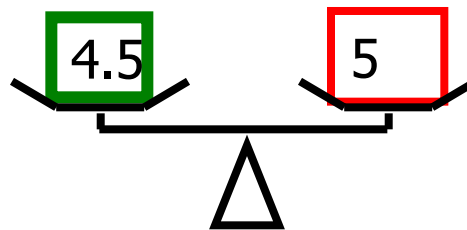
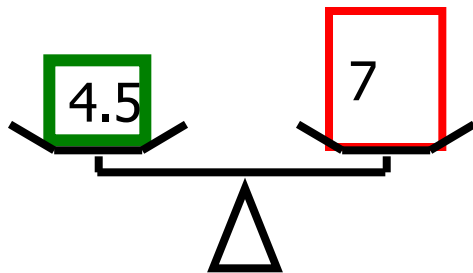


# フラッシュ型ADC 大きな冗長性



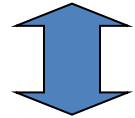
全ての重さの分銅と  
それを載せる天秤を用意

入力Vin 4.5



# フラッシュ型ADCへの見方

「フラッシュ型ADCは無駄な回路が多く賢い構成ではない」  
「6bit フラッシュADC など目をつぶっても実現できる」



「フラッシュ型ADCは偉大な構成」

- 低分解能・超高速ADCのアーキテクチャとしてフラッシュ型を超えようとして、  
(公表されてないが、まわりで)  
いくつかの研究が失敗している  
(UCLA Abidi 先生)
- 産業界で フラッシュ型は生き残っている。



# 定説へのささやかな挑戦

---

が、本当に Gray Code 入力DACは  
実現できないのか？

やってみたら 出来た！

# Gray-code入力DA変換器の検討

群馬大学理工学府 修士2年  
姜 日晨\* Gopal Adhikari 小林 春夫

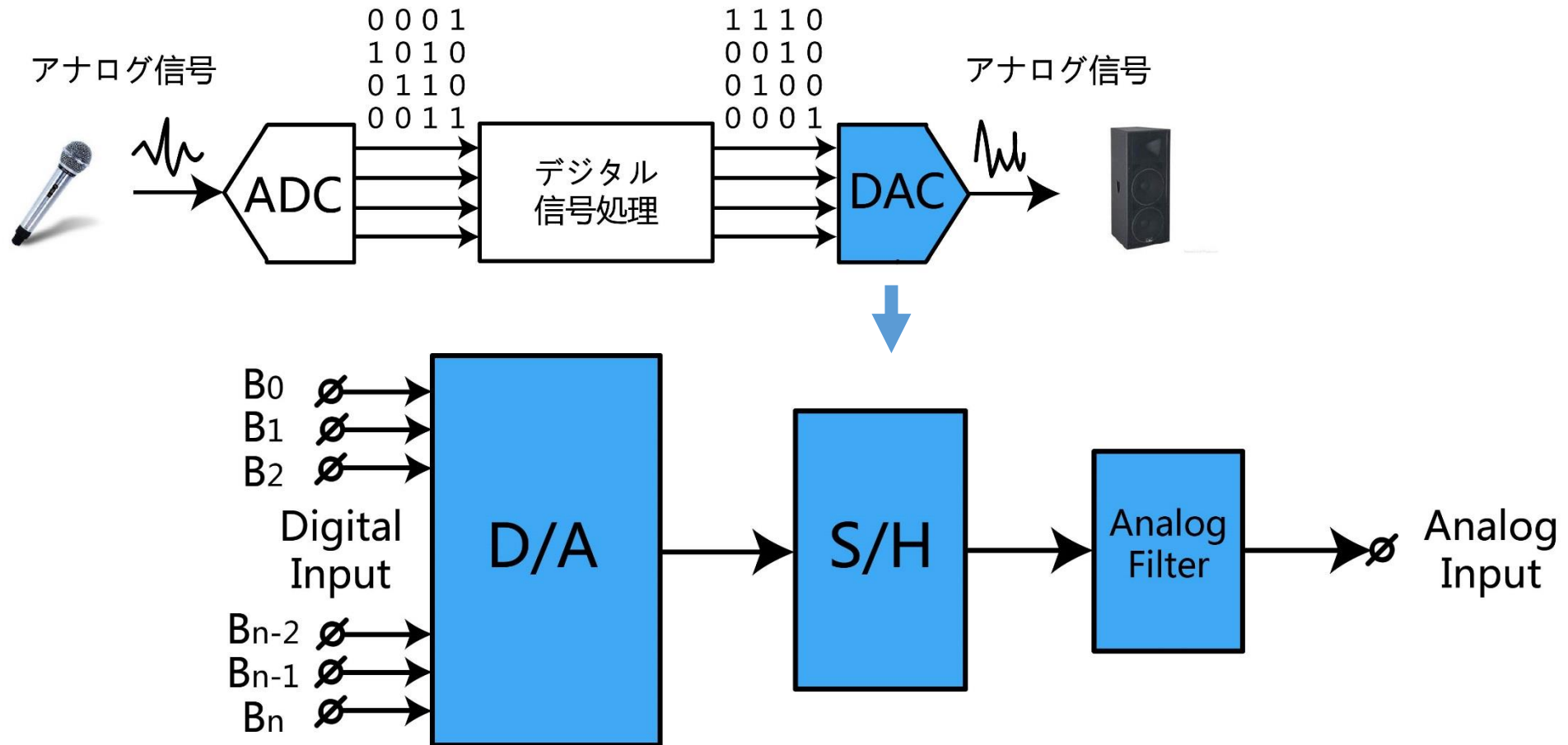
# 目次

- I. 研究背景・目的
- II. 提案するGray-code入力のDACの構成と動作
- III. SPICEによるシミュレーション検証
- IV. まとめ

# 目次

- I. 研究背景・目的
- II. 提案するGray-code入力のDACの構成と動作
- III. SPICEによるシミュレーション検証
- IV. まとめ

# デジタル-アナログコンバータ(DAC、ダック)



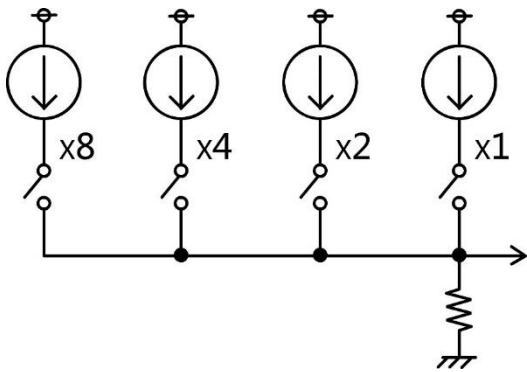
# 研究目的

Gray-codeを入力としての綺麗な構成のDACは**実現が難しい**と考えられてきた.

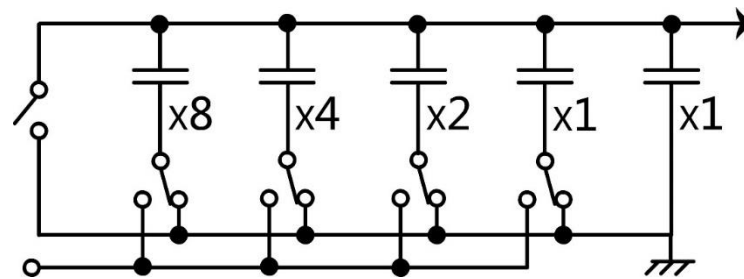


本論文では**Gray-code入力のDAC**が実現できることを示す.

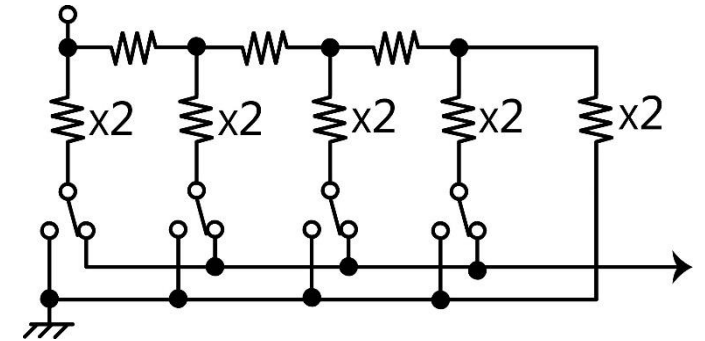
# D/A変換器の基本的なアーキテクチャ



電流源型DAC



容量型DAC



抵抗型DAC

スイッチは2進数(バイナリコード、Binary code)で駆動 → グリッチ

# グリッチ (Glitch) の影響



グラフィックディスプレイでのグリッチ



# グリッチの発生原理

| Decimal numbers | Natural Binary code |
|-----------------|---------------------|
| 0               | 0 0 0 0             |
| 1               | 0 0 0 1             |
| 2               | 0 0 1 0             |
| 3               | 0 0 1 1             |
| 4               | 0 1 0 0             |
| 5               | 0 1 0 1             |
| 6               | 0 1 1 0             |
| 7               | 0 1 1 1             |
| 8               | 1 0 0 0             |
| 9               | 1 0 0 1             |
| 10              | 1 0 1 0             |
| 11              | 1 0 1 1             |
| 12              | 1 1 0 0             |
| 13              | 1 1 0 1             |
| 14              | 1 1 1 0             |
| 15              | 1 1 1 1             |

7→8  
の時

0111→0110→0100→**0000**→1000

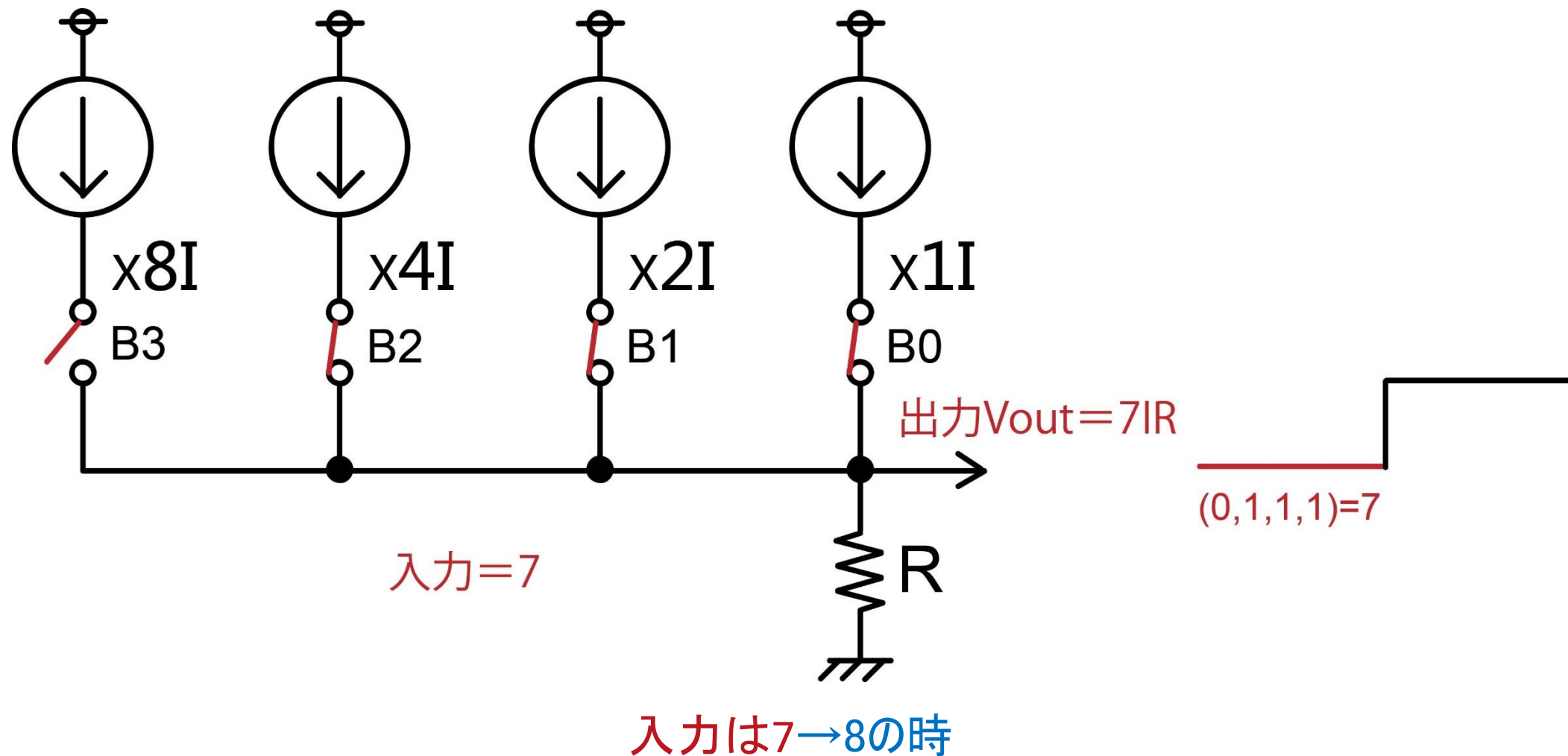
8→7  
の時

1000→1001→1011→**1111**→0111

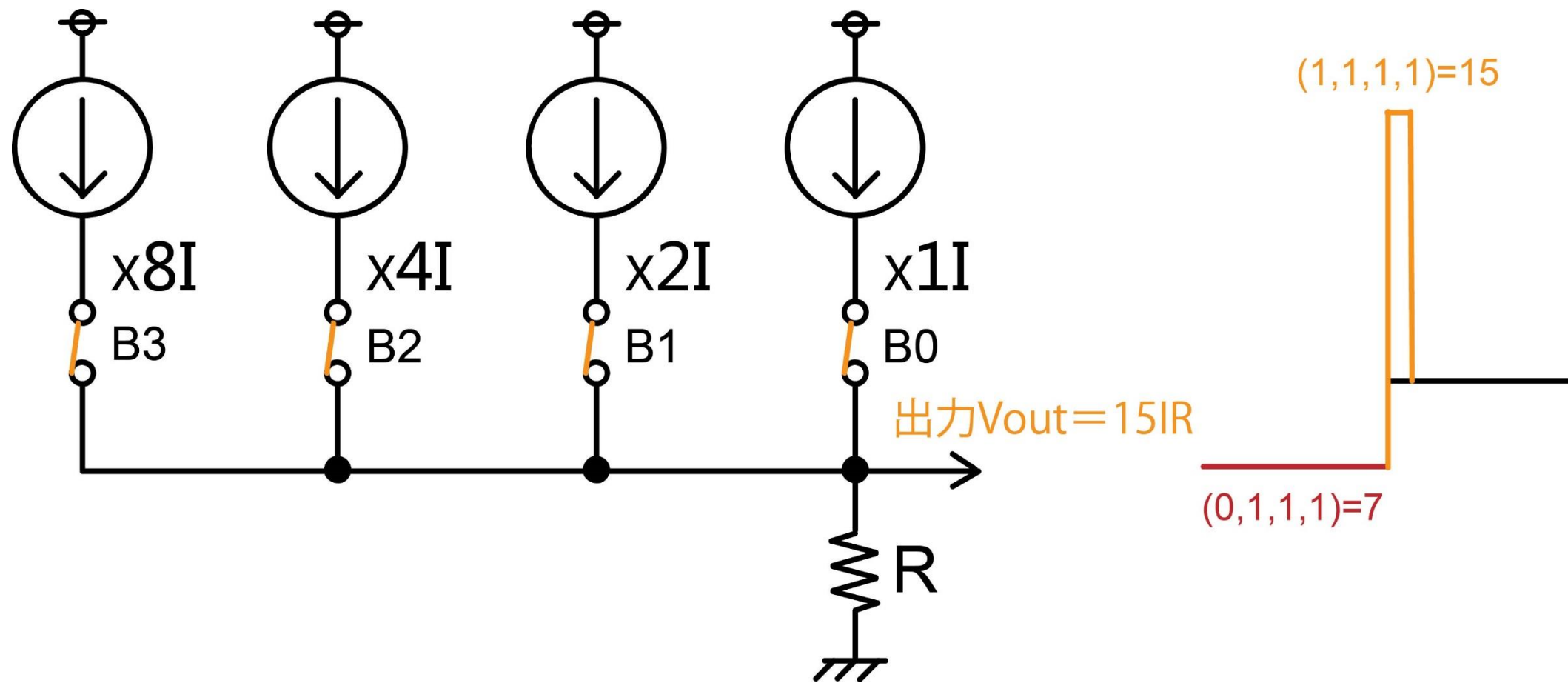
最上位ビット(MSB)が変化(中央値の付近)

ワーストケース

# スイッチ切り替えタイミング

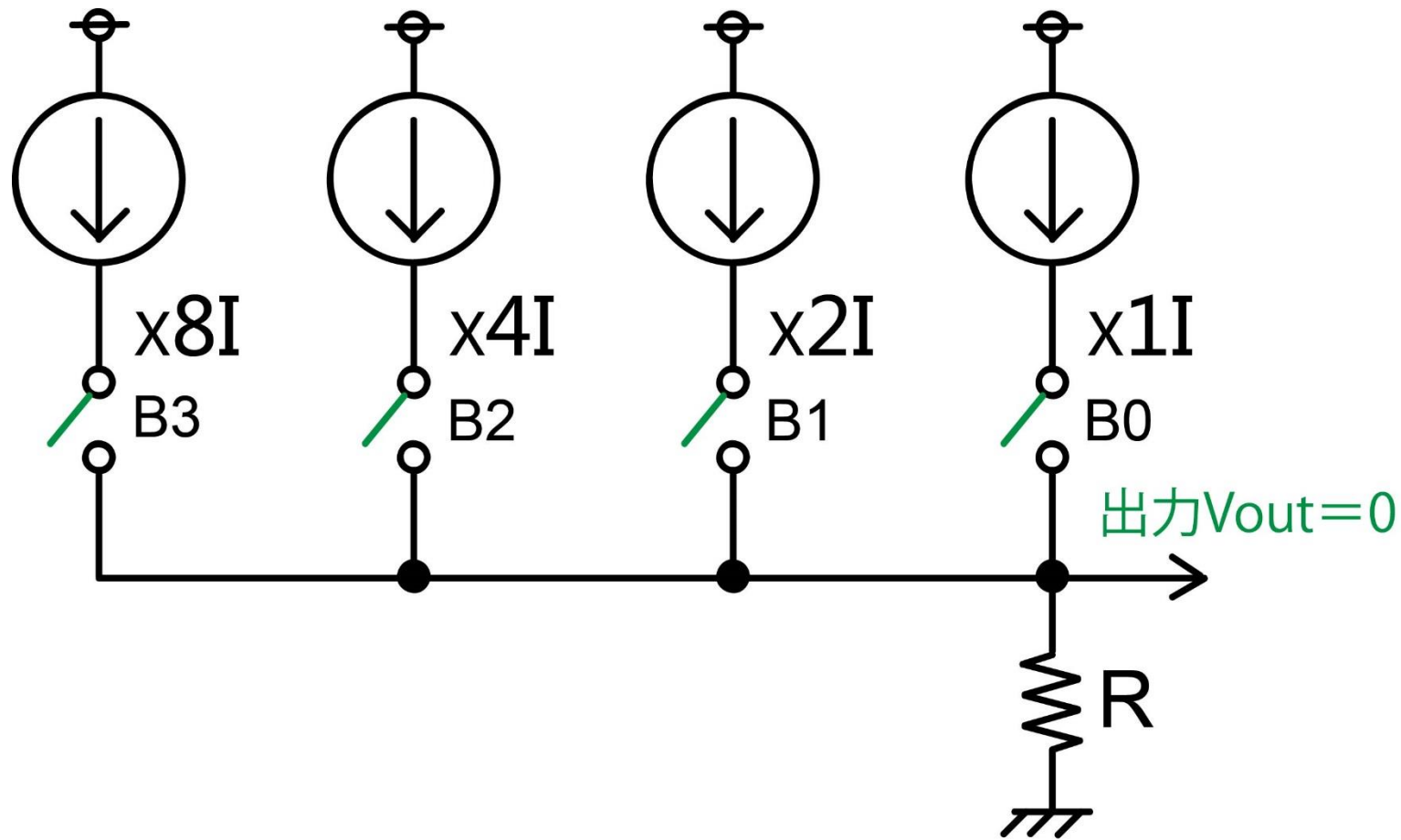


# スイッチ切り替えタイミング その 1

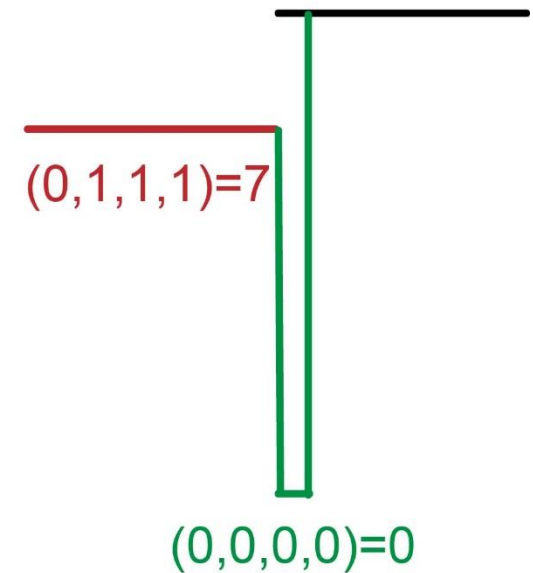


B3が最初にスイッチングすると

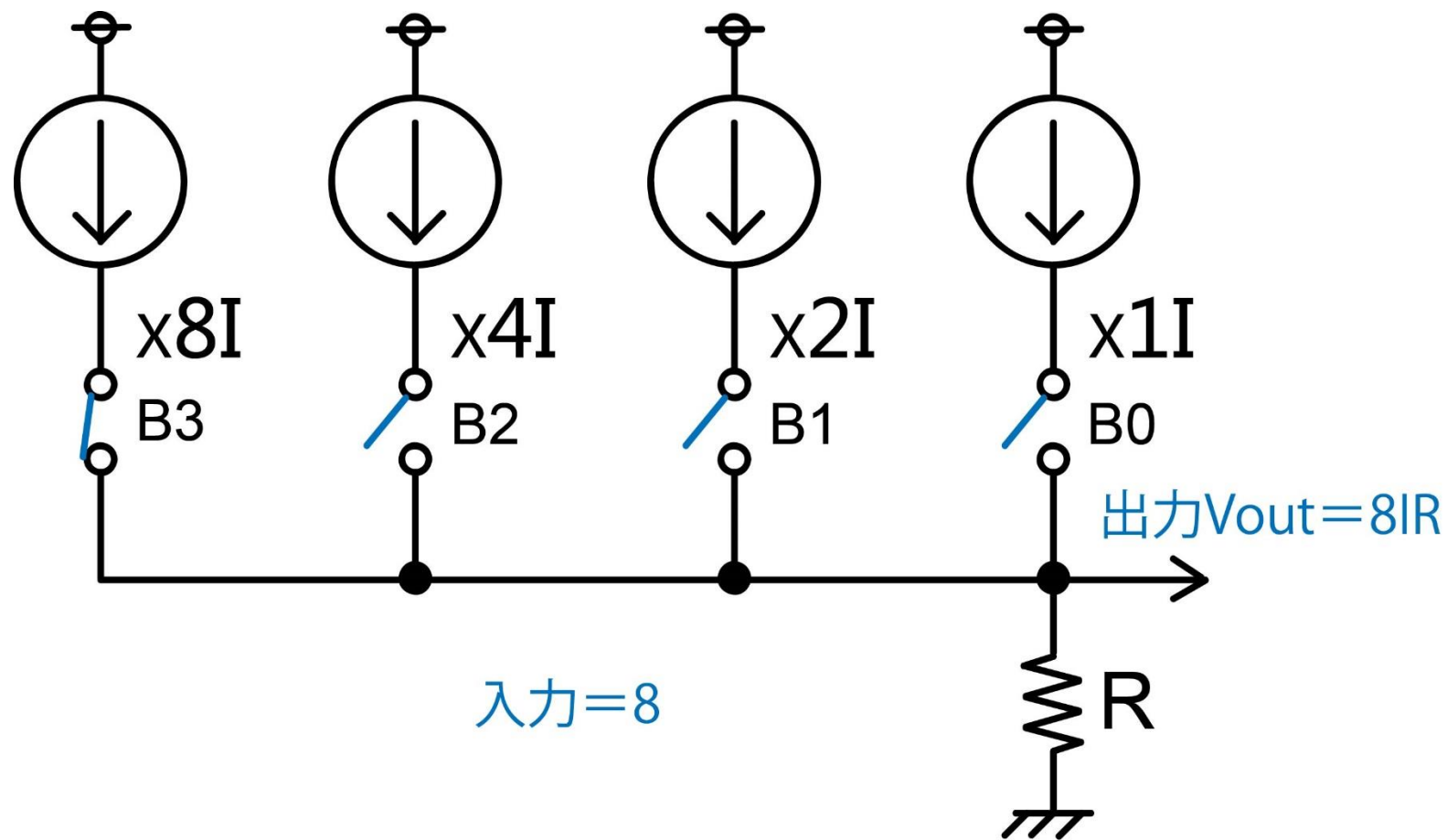
# スイッチ切り替えタイミング その 1



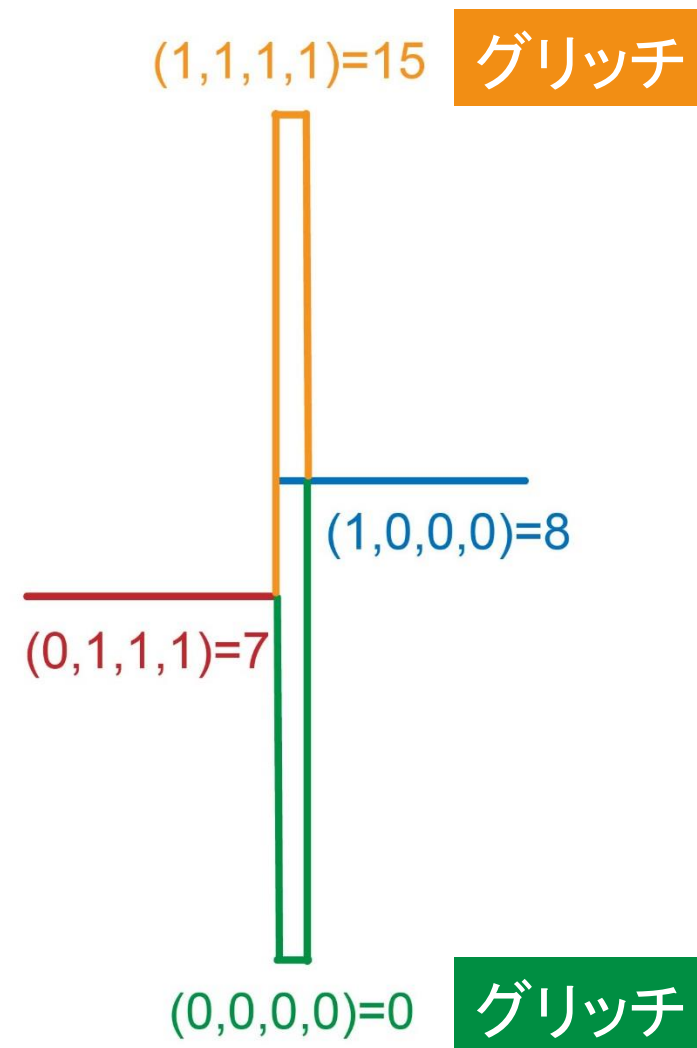
B3が最後にスイッチングすると



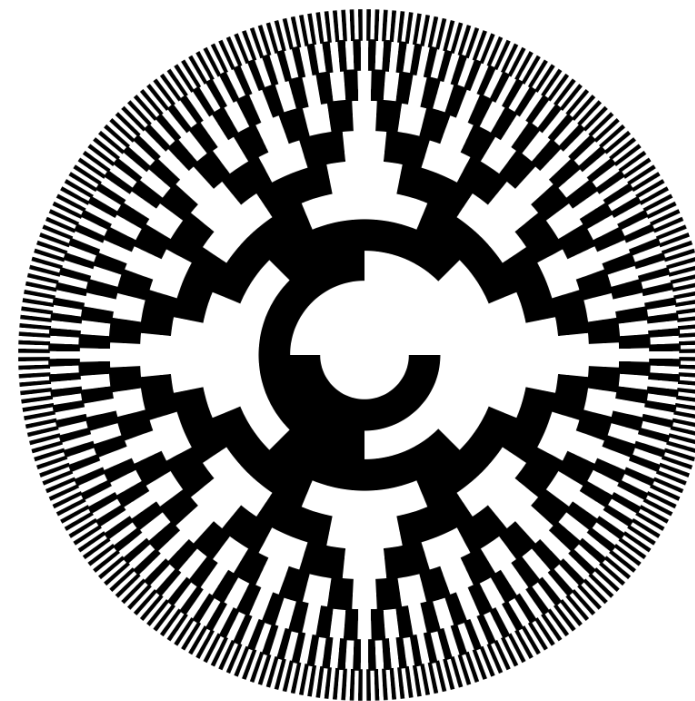
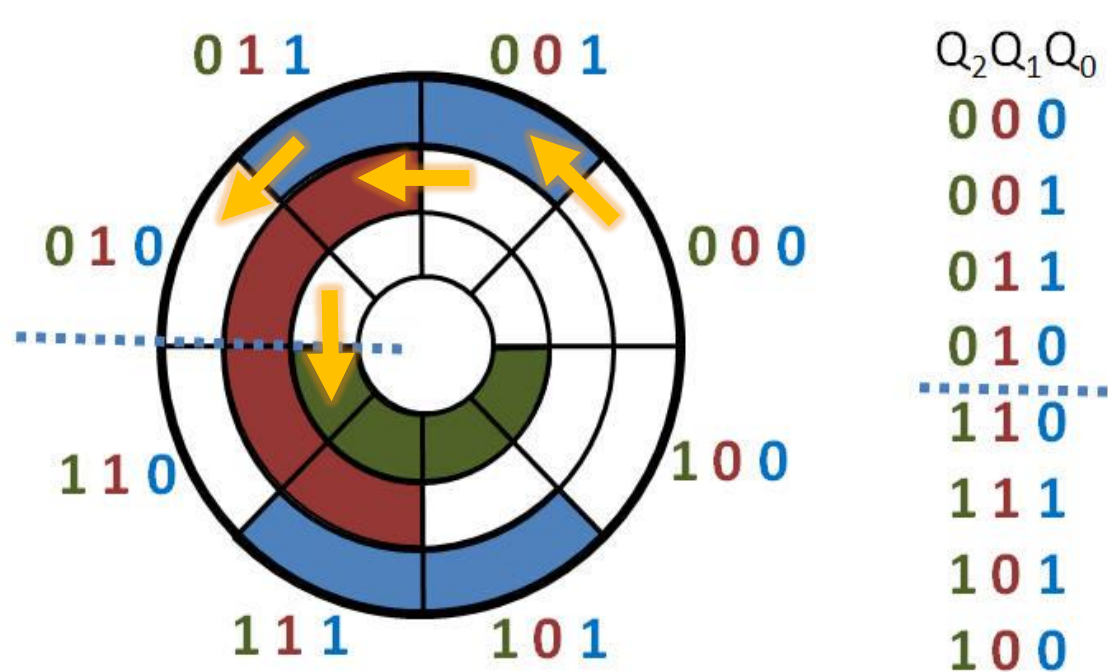
# スイッチングスキュー → グリッチ



入力は7→8の時



# グレイコード (Gray-code)



**グレイコード:** 前後に隣接する符号間のハミング距離が必ず1

ベル研究所のフランク・グレイが1947年の特許出願書で最初に使用した。

# グレイコードD/A変換器

Binary code と Gray code の変換が容易 (EXOR)  
( $G_n = B_{n+1} \oplus B_n$ )

Gray code 入力のDA変換器



グリッチが小さくできる

Binary code: 0111→0110→0100→0000→1000

Gray code: 0100→1100

| Decimal numbers | Natural Binary code | Gray code |
|-----------------|---------------------|-----------|
| 0               | 0 0 0 0             | 0 0 0 0   |
| 1               | 0 0 0 1             | 0 0 0 1   |
| 2               | 0 0 1 0             | 0 0 1 1   |
| 3               | 0 0 1 1             | 0 0 1 0   |
| 4               | 0 1 0 0             | 0 1 1 0   |
| 5               | 0 1 0 1             | 0 1 1 1   |
| 6               | 0 1 1 0             | 0 1 0 1   |
| 7               | 0 1 1 1             | 0 1 0 0   |
| 8               | 1 0 0 0             | 1 1 0 0   |
| 9               | 1 0 0 1             | 1 1 0 1   |
| 10              | 1 0 1 0             | 1 1 1 1   |
| 11              | 1 0 1 1             | 1 1 1 0   |
| 12              | 1 1 0 0             | 1 0 1 0   |
| 13              | 1 1 0 1             | 1 0 1 1   |
| 14              | 1 1 1 0             | 1 0 0 1   |
| 15              | 1 1 1 1             | 1 0 0 0   |

# 目次

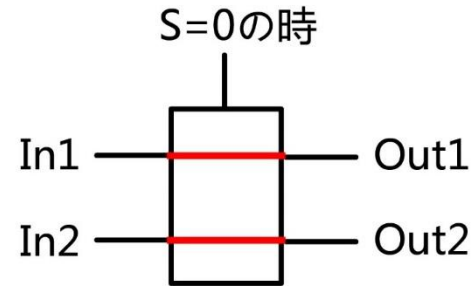
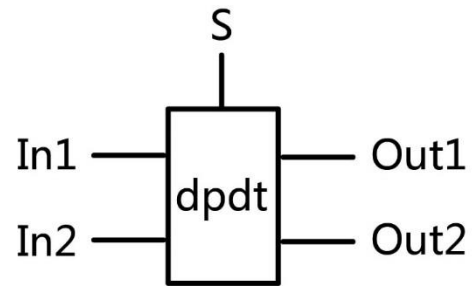
- I. 研究背景・目的
- II. 提案するGray-code入力のDACの構成と動作
- III. SPICEによるシミュレーション検証
- IV. まとめ



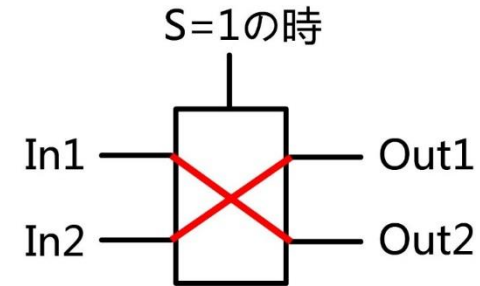
# 提案するGray-code入力のDACの構成と動作

1. Gray-code入力の電流源型DAC (I-DAC)
2. Gray-code入力の容量型DAC (C-DAC)
3. Gray-code入力の電圧加算型DAC (V-DAC)

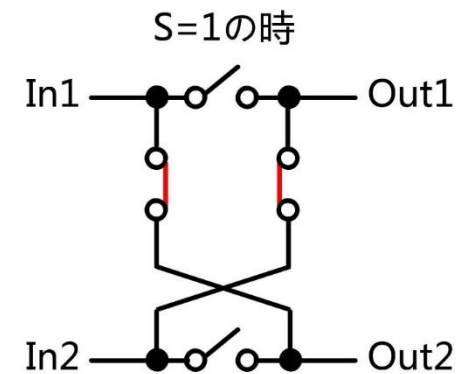
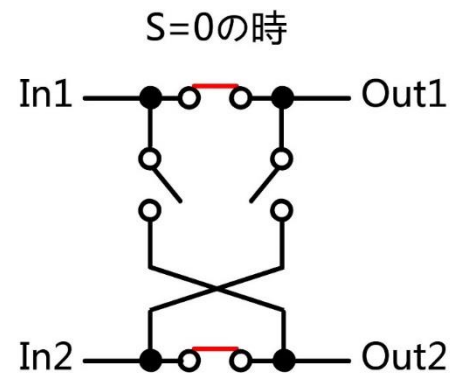
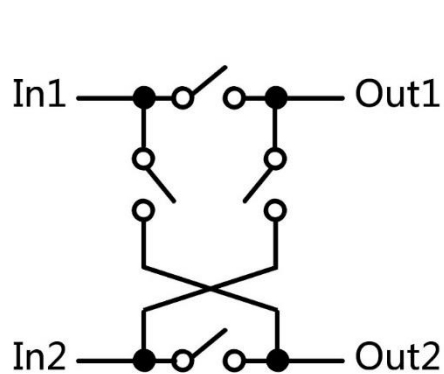
# 電流/電圧スイッチマトリックス



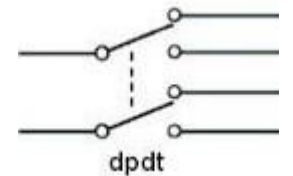
平行接続



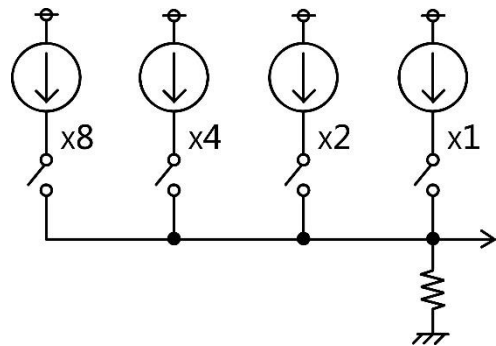
クロス接続



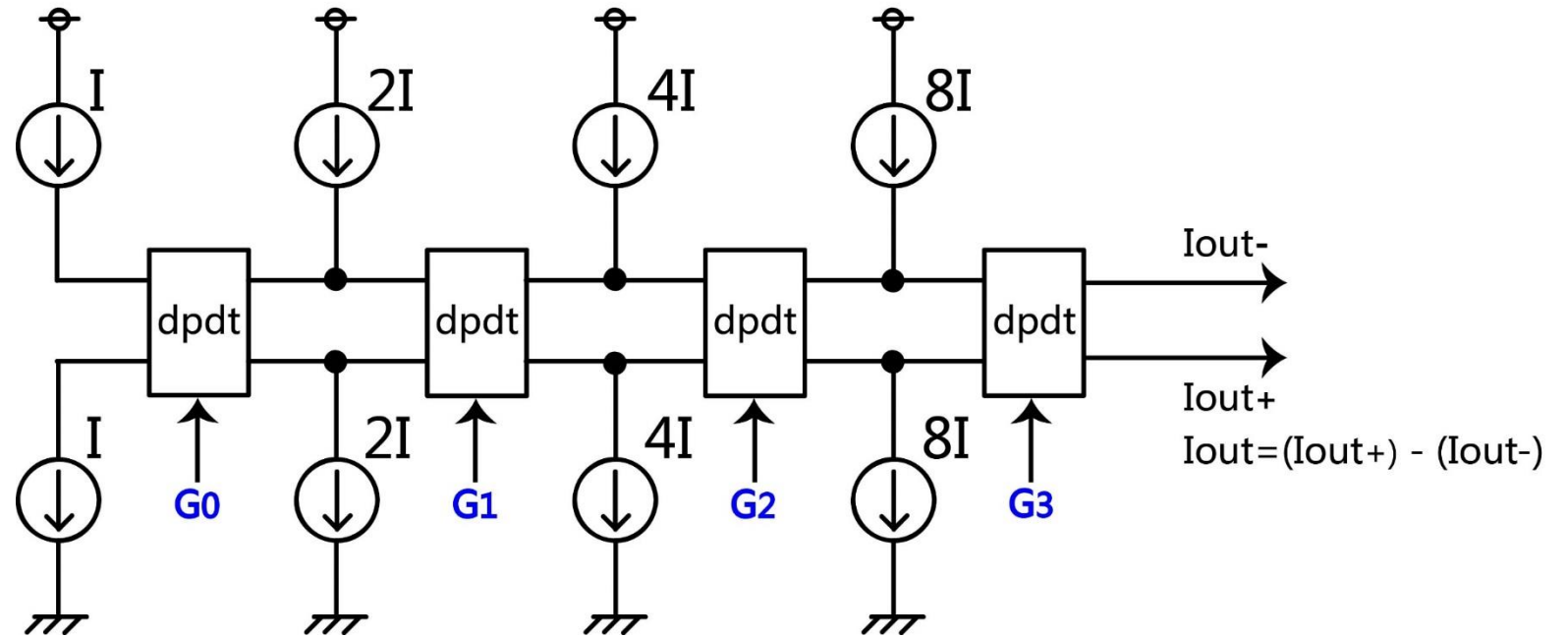
double-pole double-throw (dpdt, 双極双投) スイッチで実現



# 1. Gray-code入力の電流源型DAC

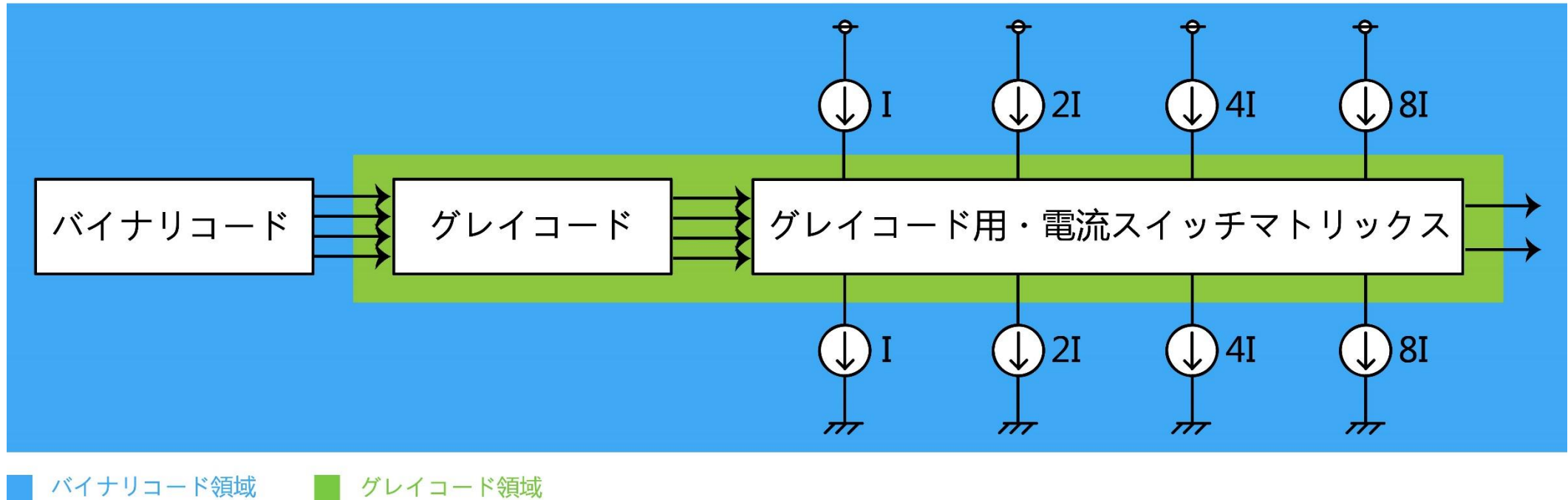


従来の電流源型DAC



Gray-code入力の電流源型DAC

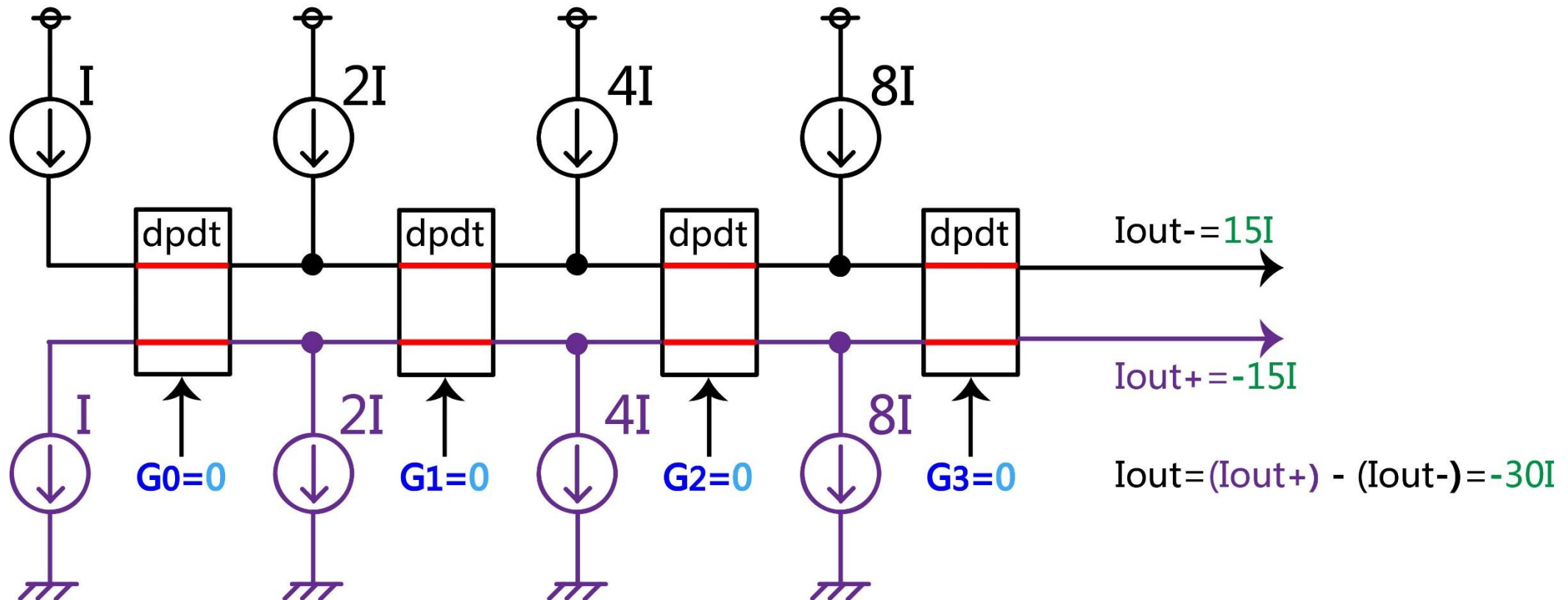
# コード変換



バイナリの電流源順番に対応できるGray-code入力

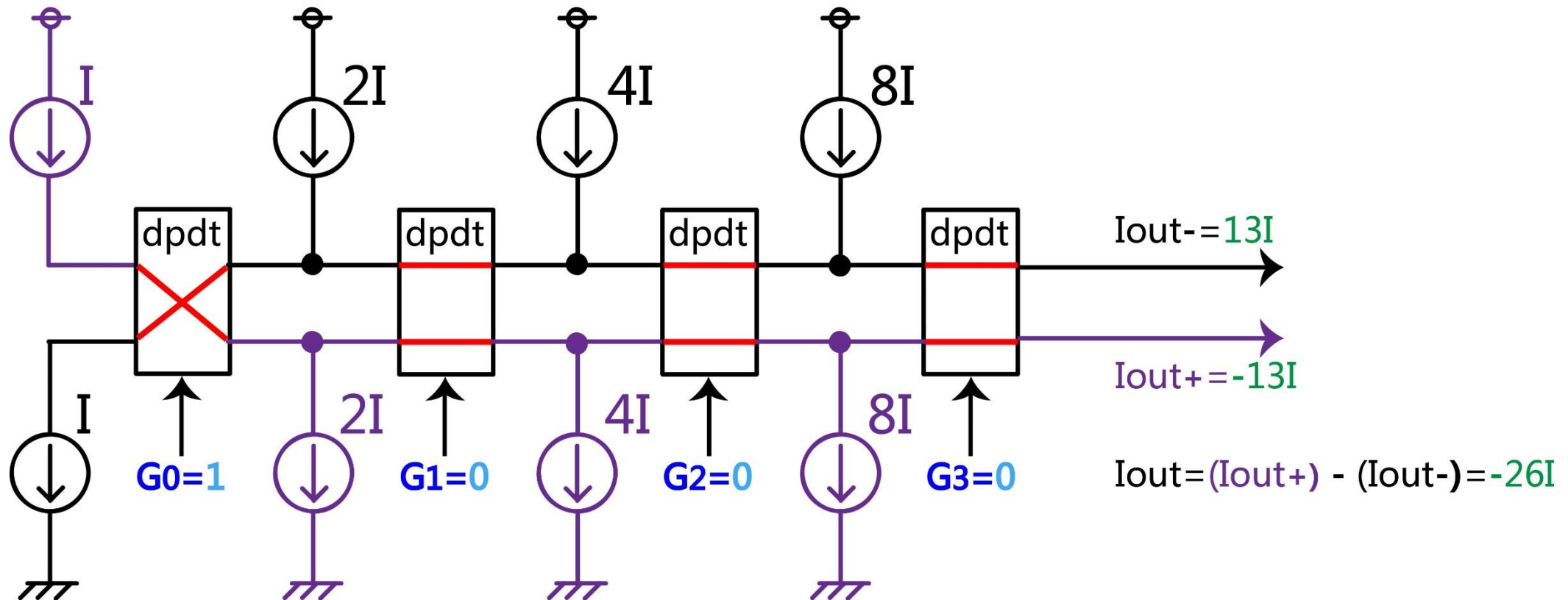
# Gray-code入力のI-DACの動作 (データ=0 の場合)

eg. Data=0



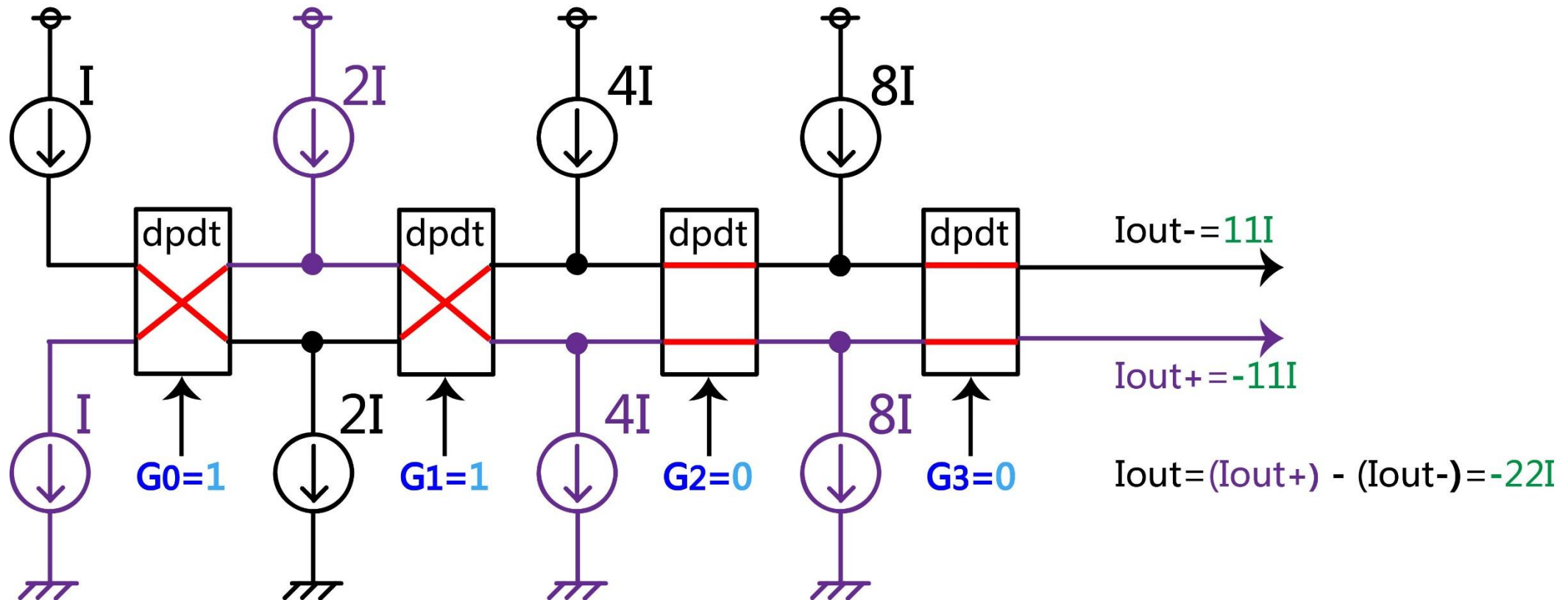
# Gray-code入力のI-DACの動作 (データ=1 の場合)

eg. Data=1



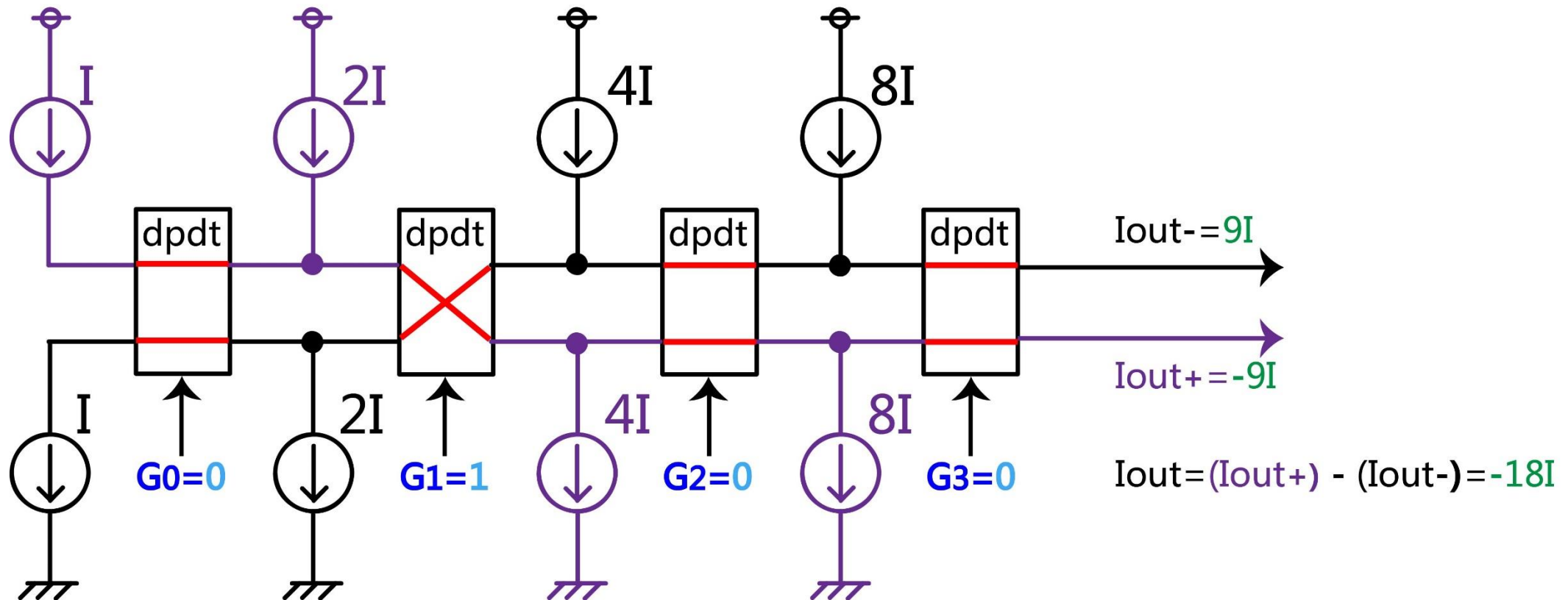
# Gray-code入力のI-DACの動作 (データ=2 の場合)

eg. Data=2



# Gray-code入力のI-DACの動作 (データ=3 の場合)

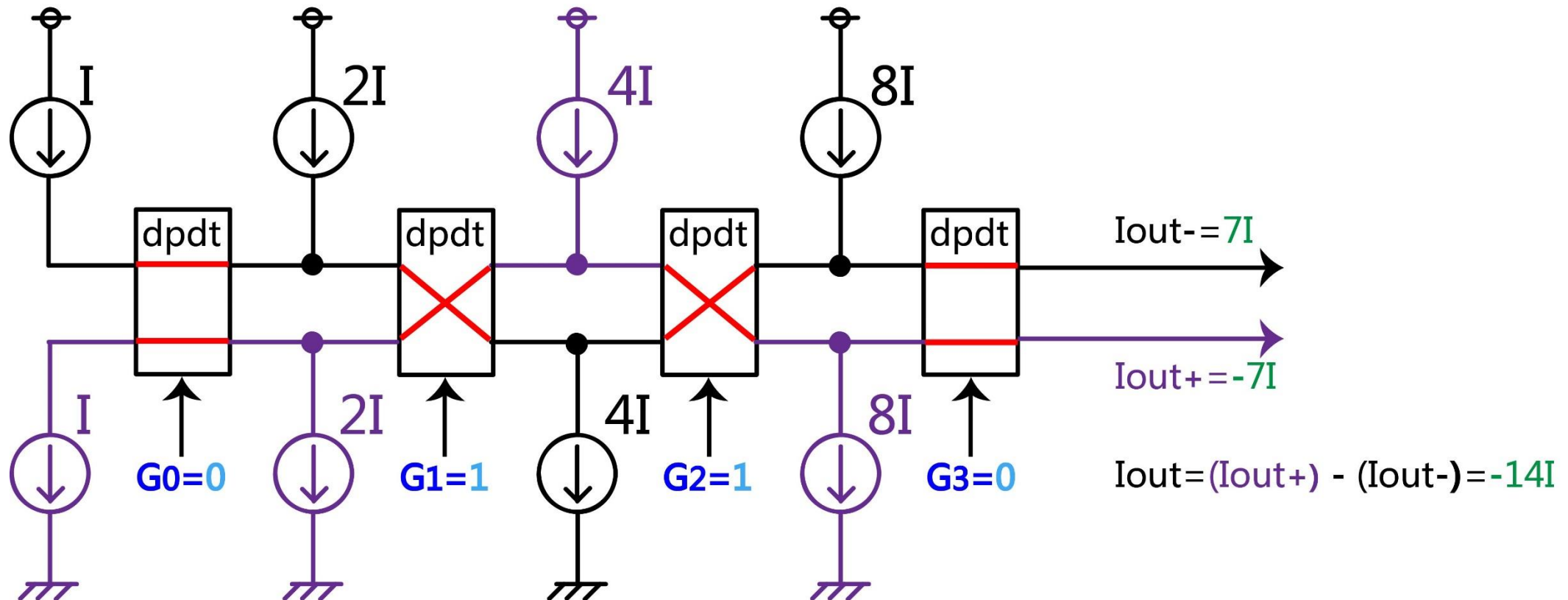
eg. Data=3





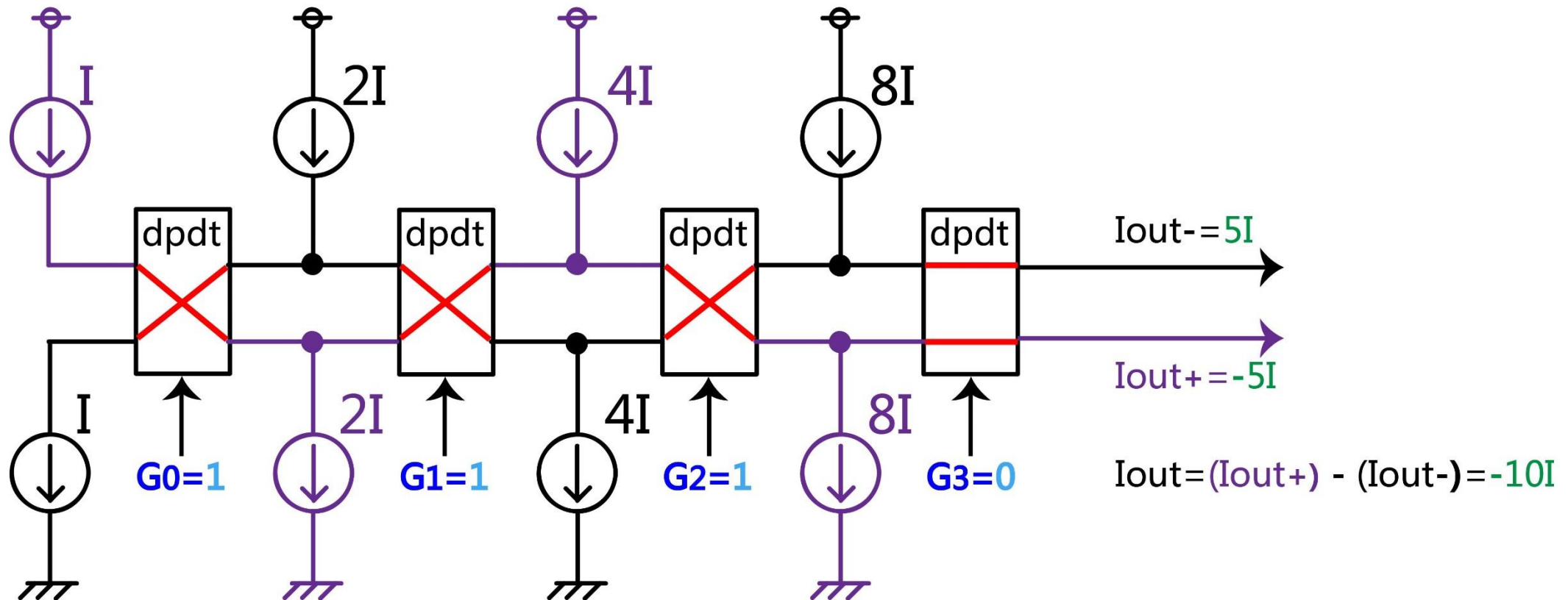
# Gray-code入力のI-DACの動作 (データ=4 の場合)

eg. Data=4



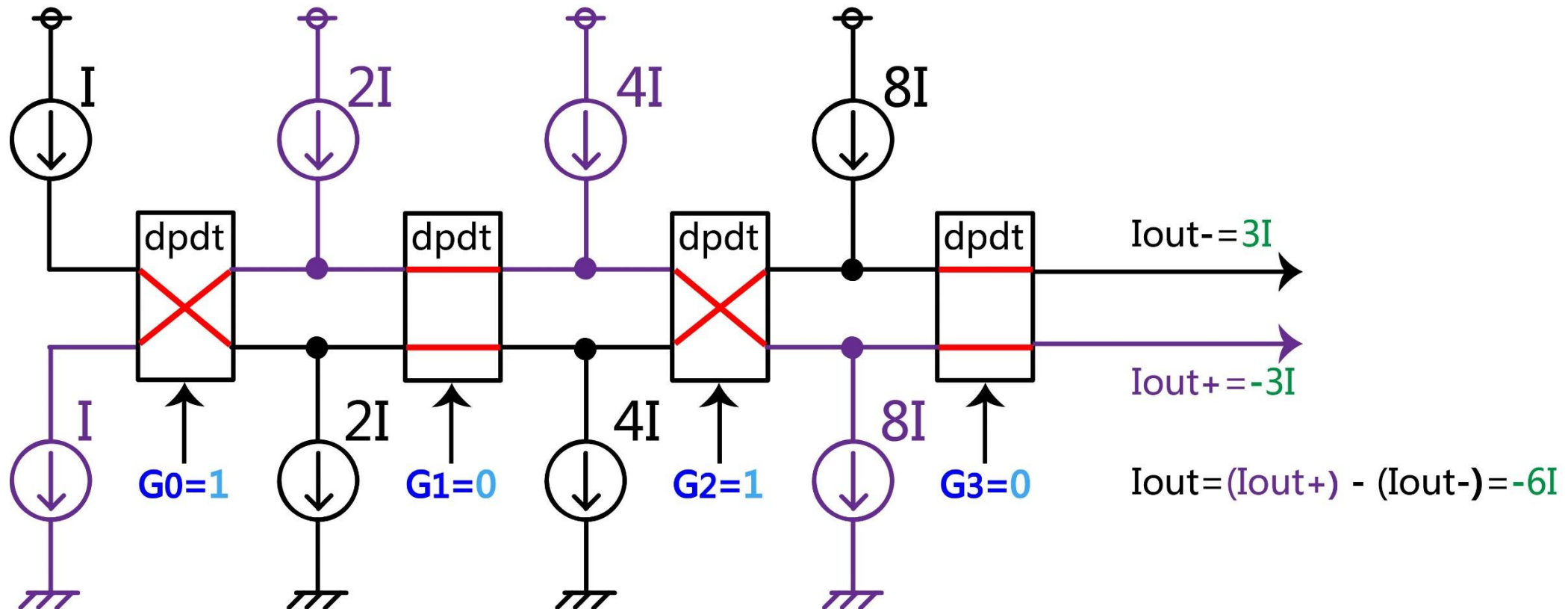
# Gray-code入力のI-DACの動作 (データ=5 の場合)

eg. Data=5



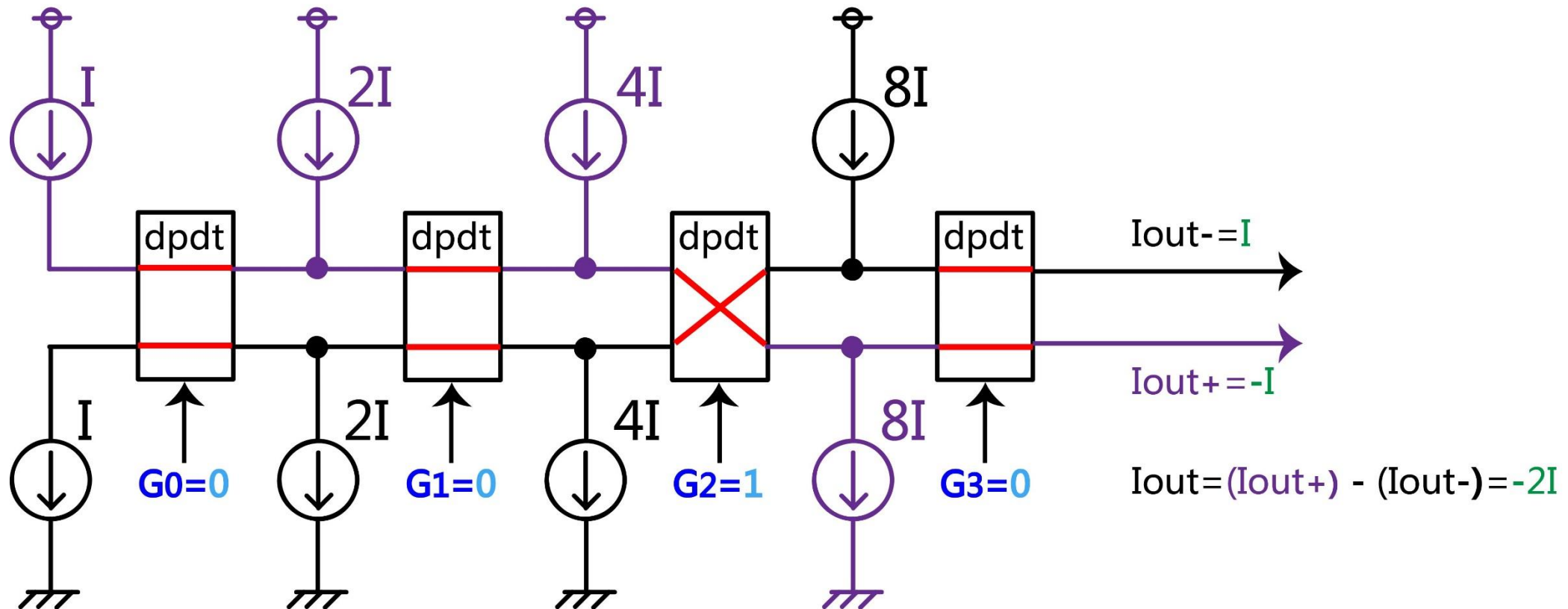
# Gray-code入力のI-DACの動作 (データ=6 の場合)

eg. Data=6



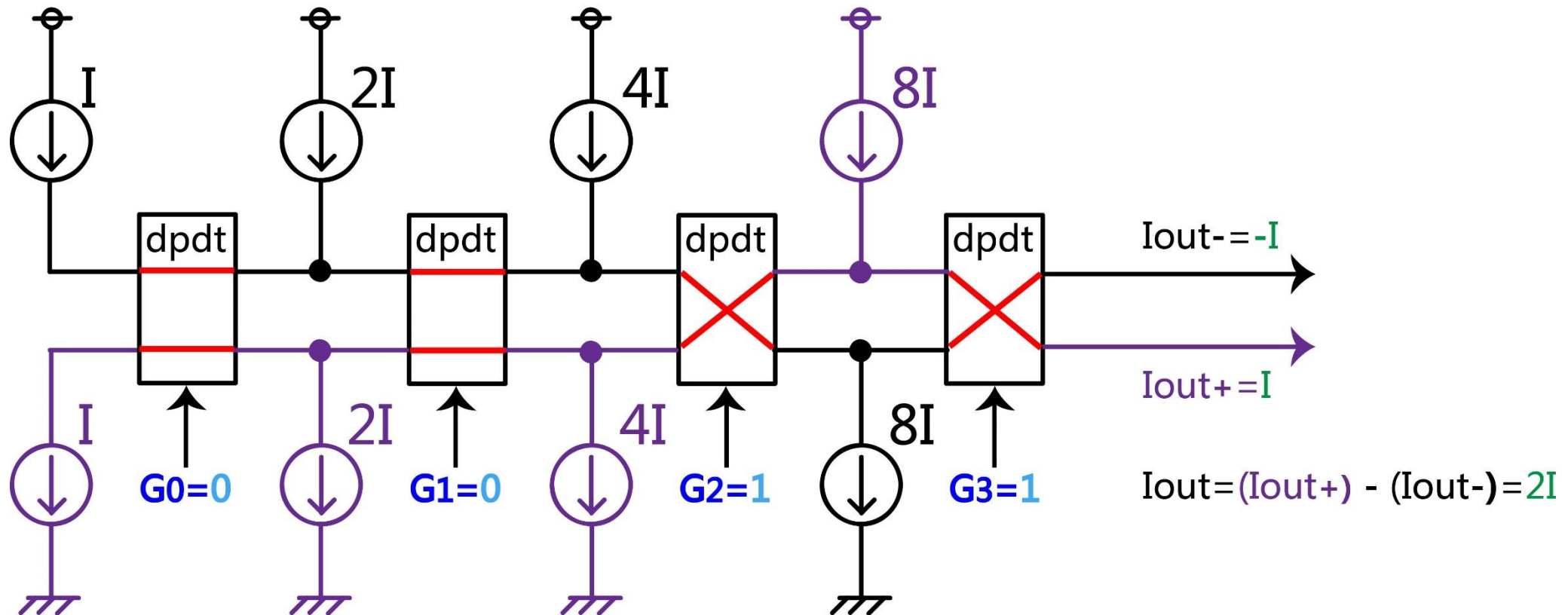
# Gray-code入力のI-DACの動作 (データ=7 の場合)

eg. Data=7



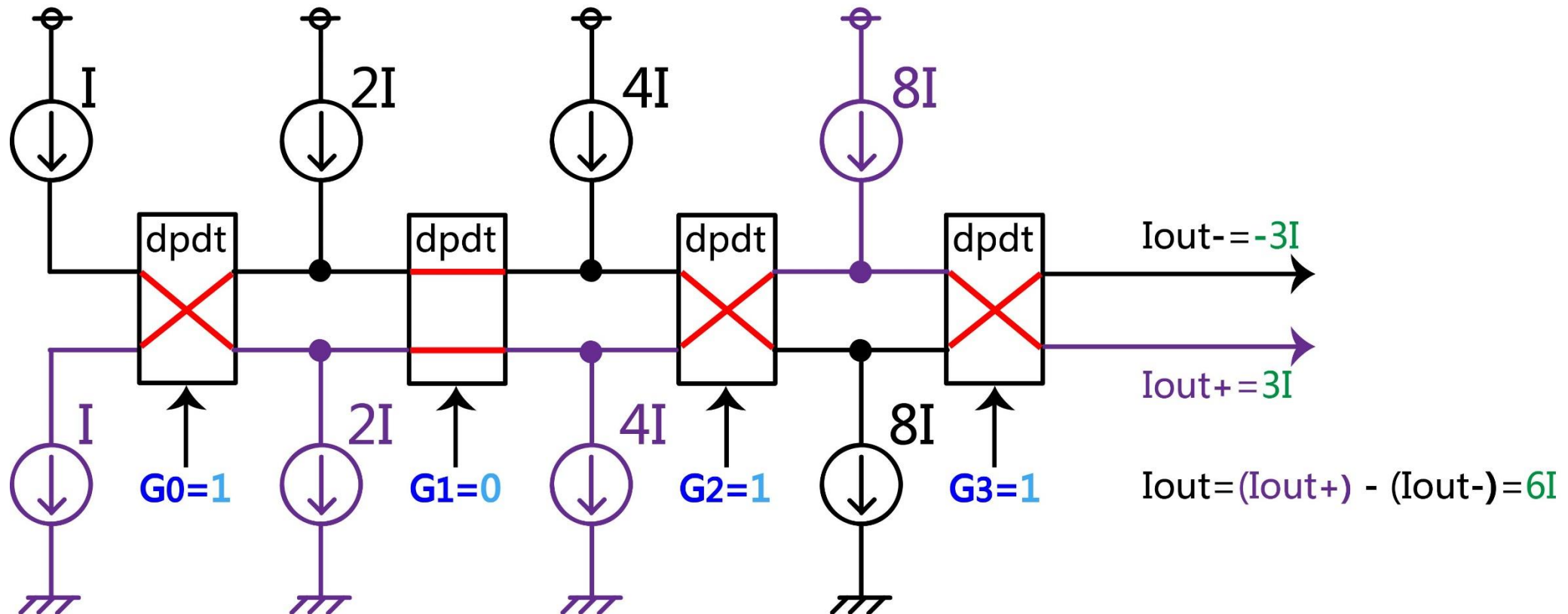
# Gray-code入力のI-DACの動作 (データ=8 の場合)

eg. Data=8



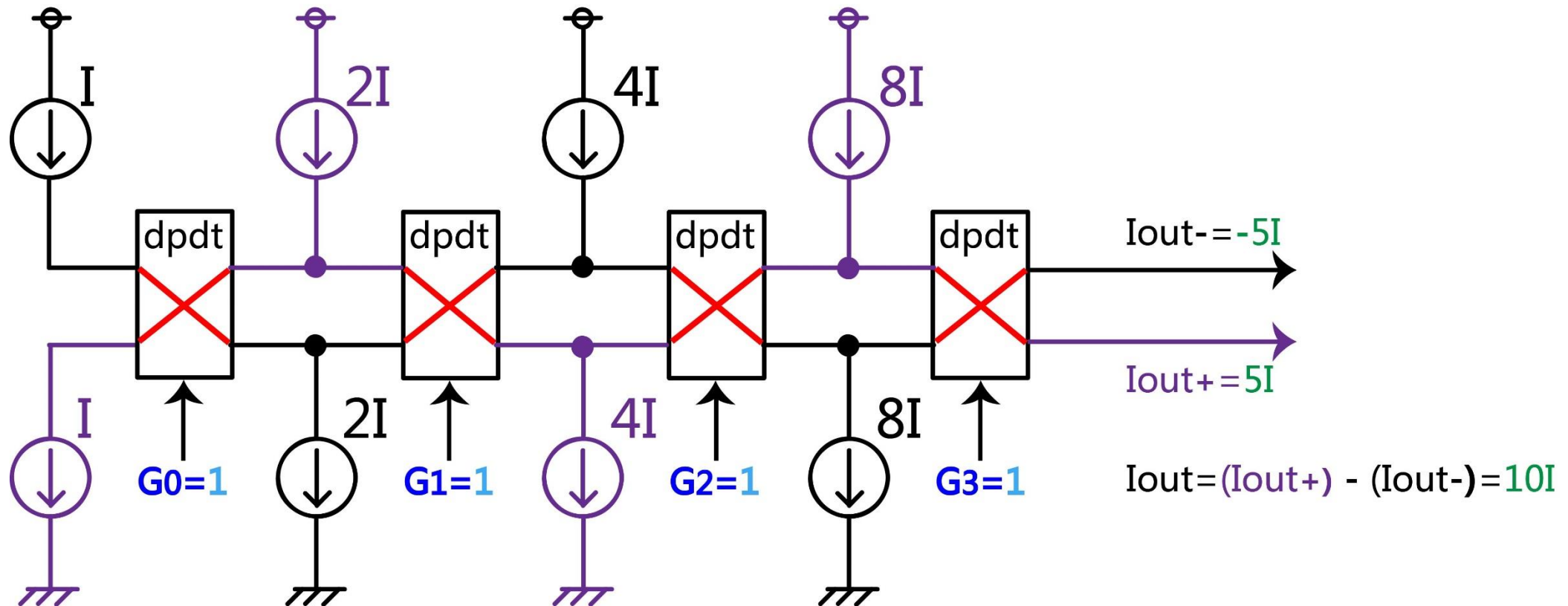
# Gray-code入力のI-DACの動作 (データ=9 の場合)

eg. Data=9



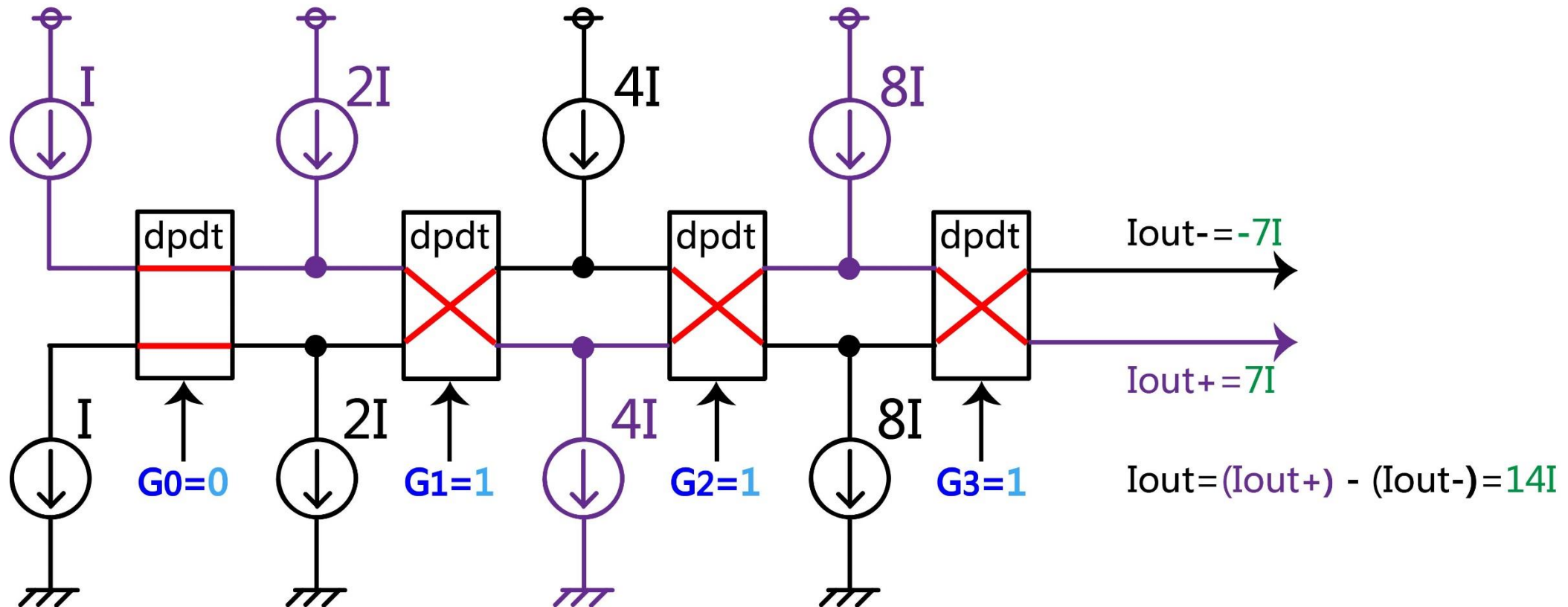
# Gray-code入力のI-DACの動作 (データ=10 の場合)

eg. Data=10



# Gray-code入力のI-DACの動作 (データ=11 の場合)

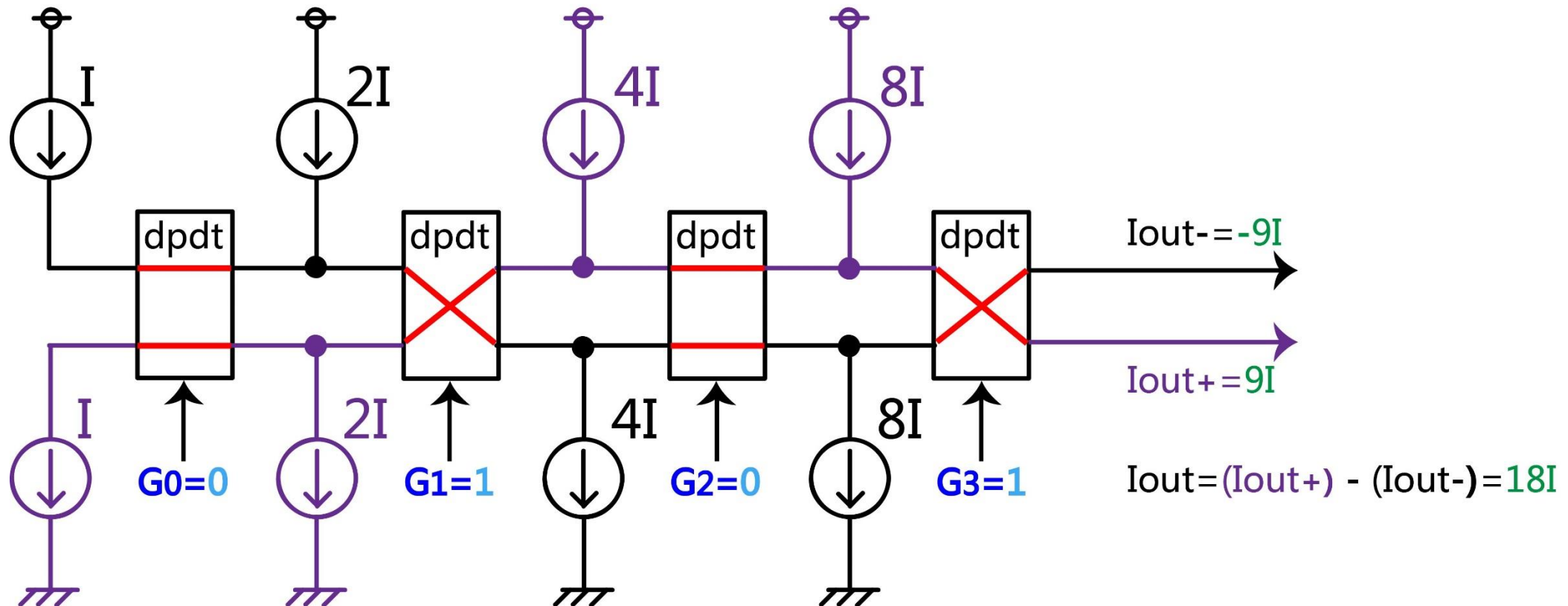
eg. Data=11





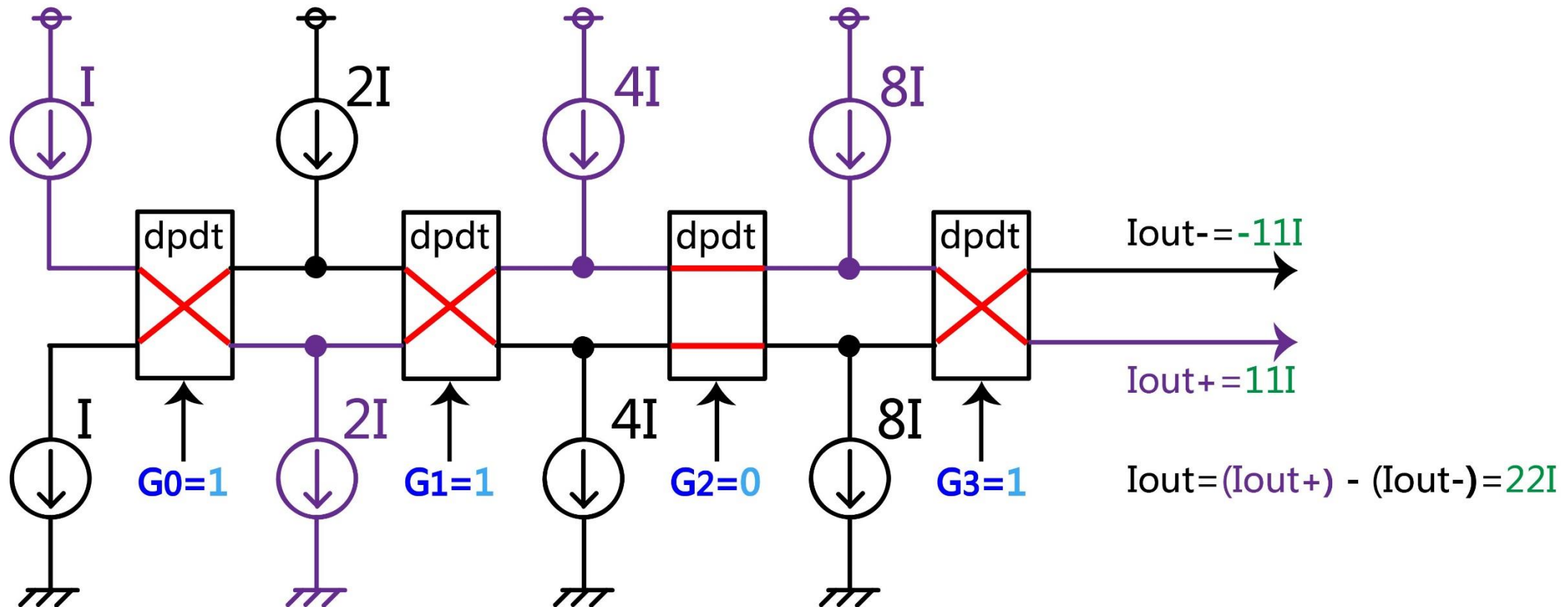
# Gray-code入力のI-DACの動作 (データ=12 の場合)

eg. Data=12



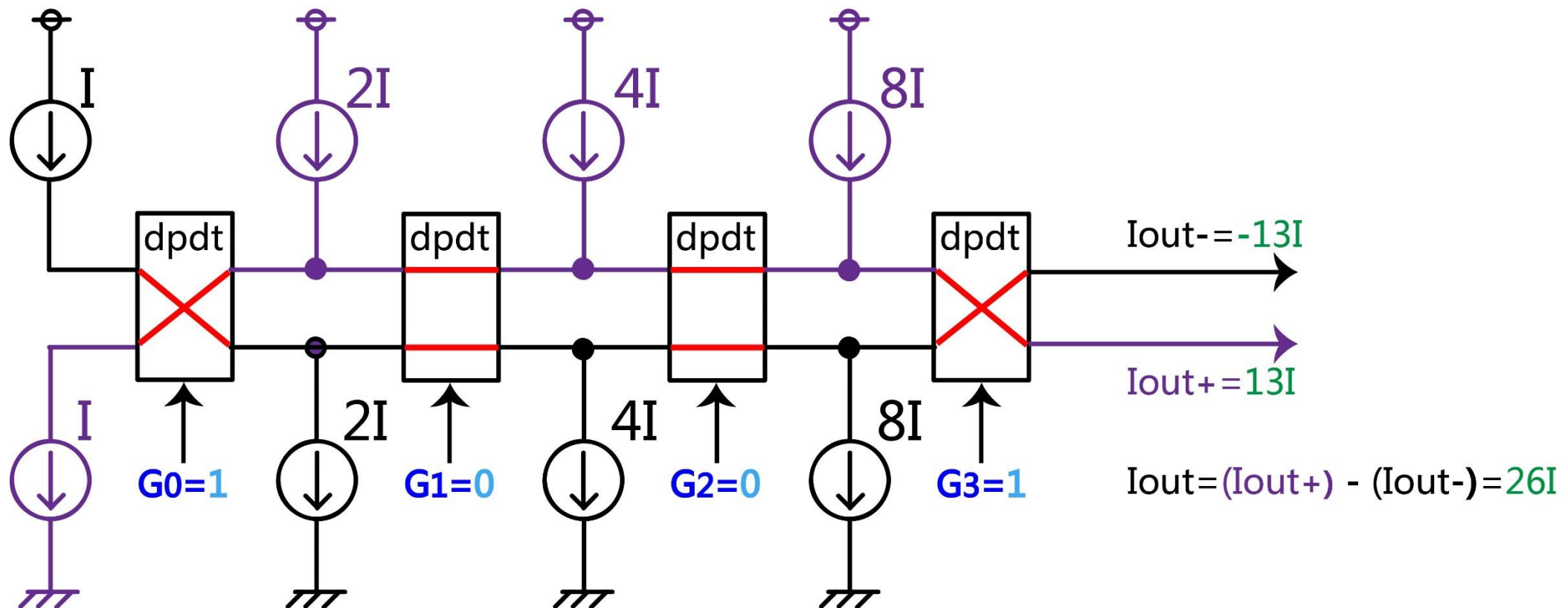
# Gray-code入力のI-DACの動作 (データ=13 の場合)

eg. Data=13



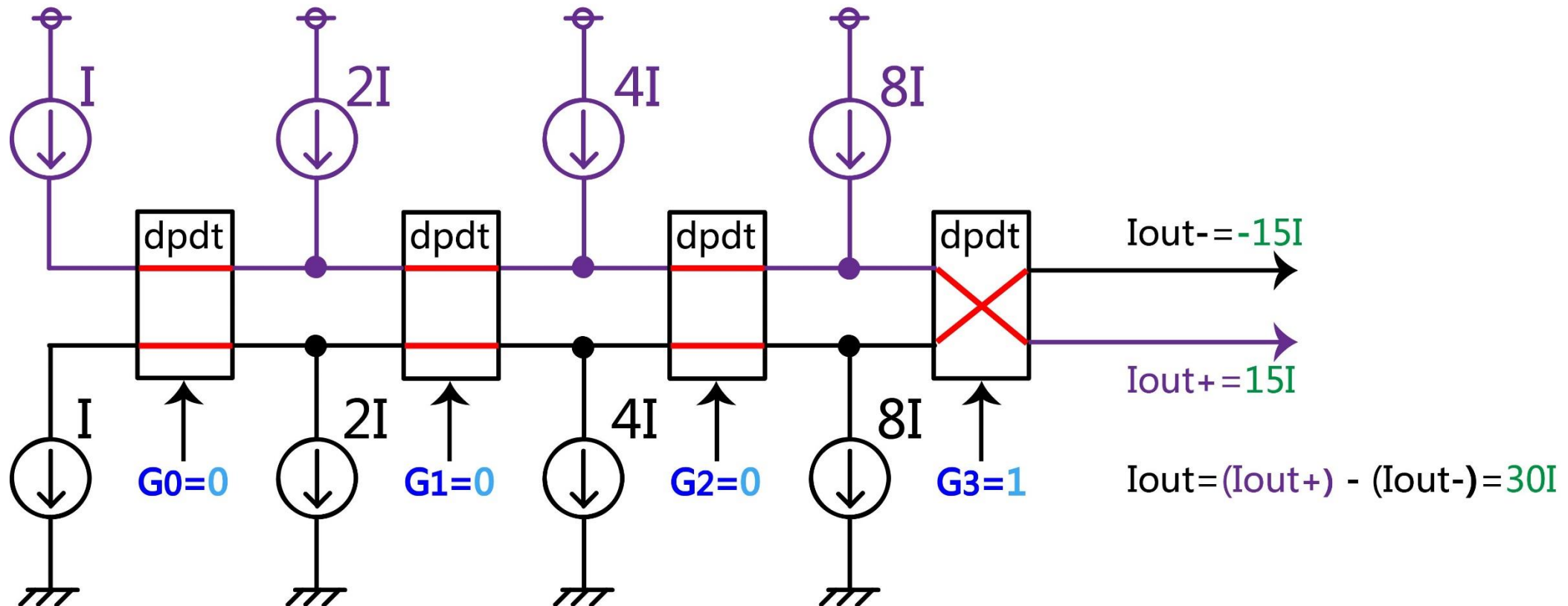
# Gray-code入力のI-DACの動作 (データ=14 の場合)

eg. Data=14

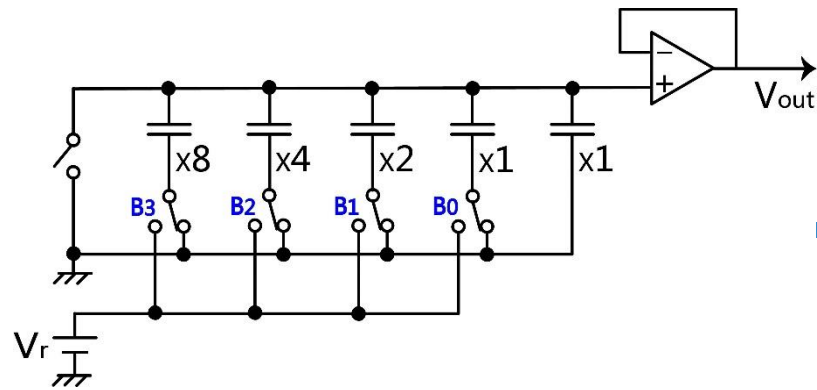


# Gray-code入力のI-DACの動作 (データ=15 の場合)

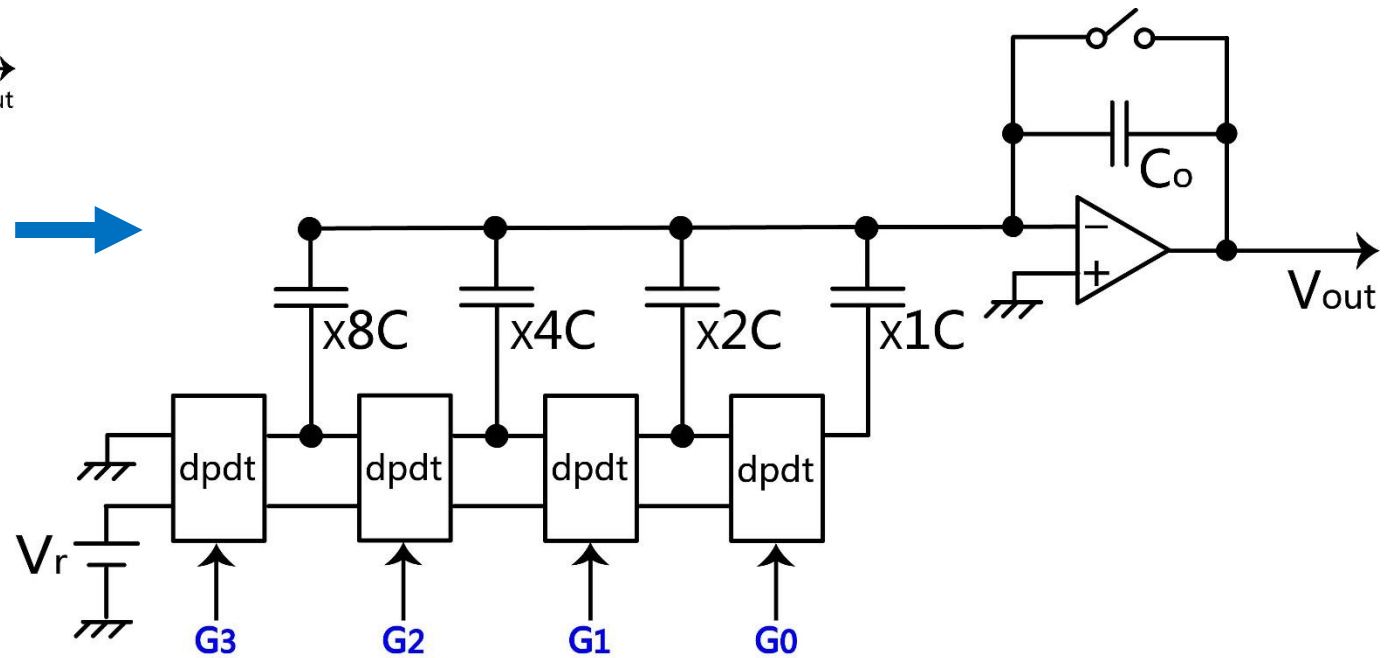
eg. Data=15



## 2. Gray-code入力の容量型DAC



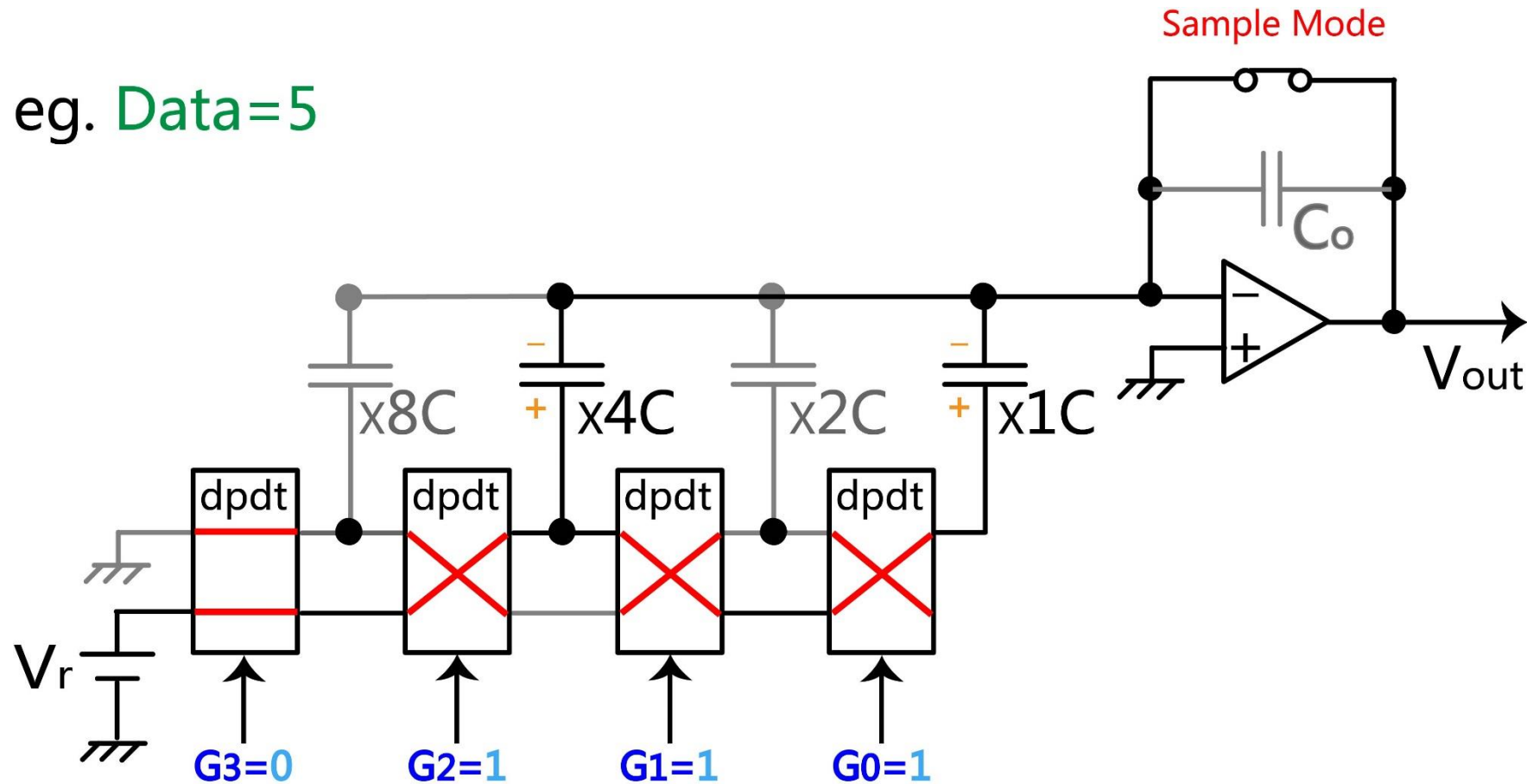
従来の容量型DAC



Gray-code入力の容量型DAC

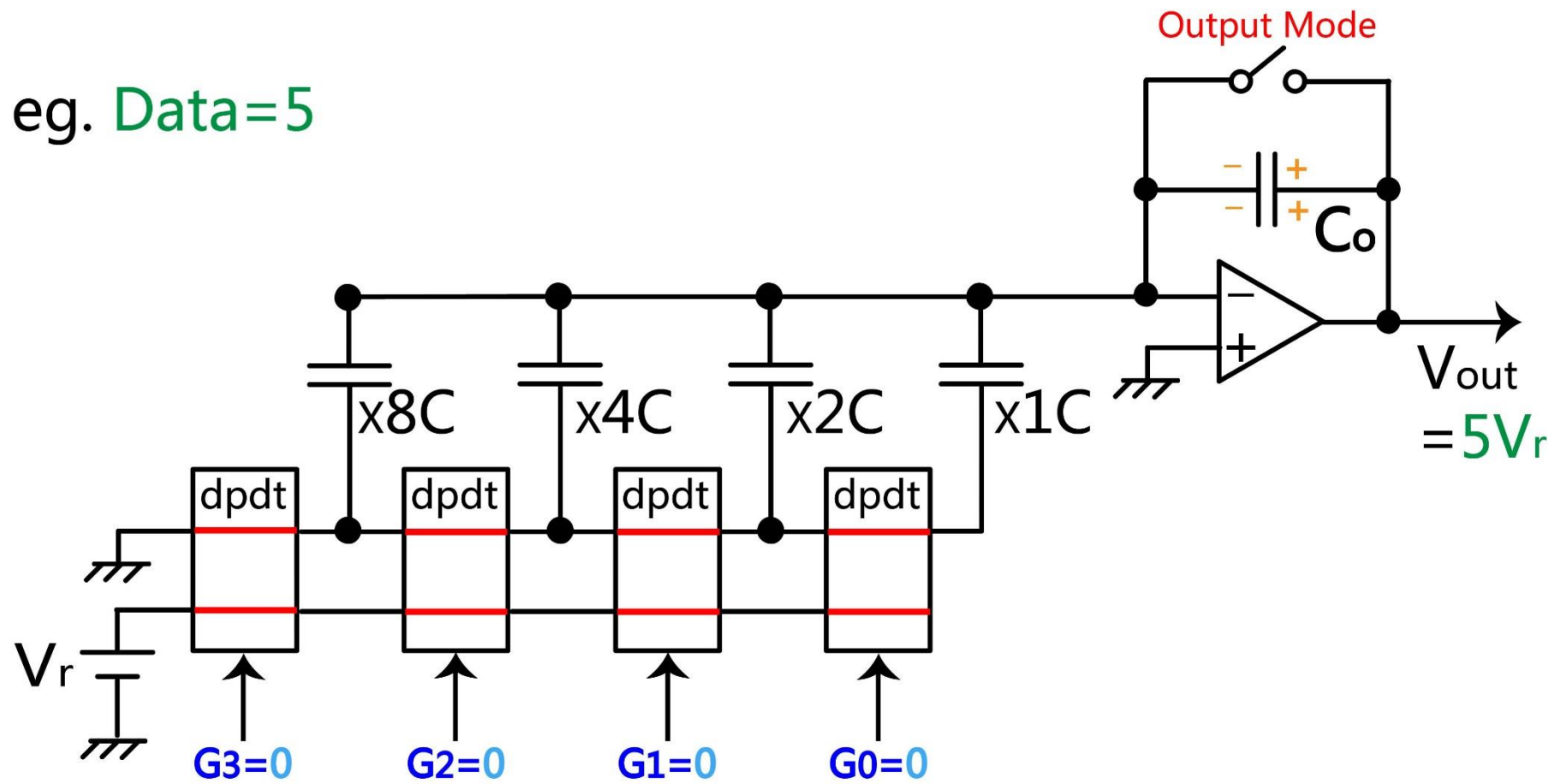
# Gray-code入力のC-DACの動作 (データ=5の場合)

eg. Data=5

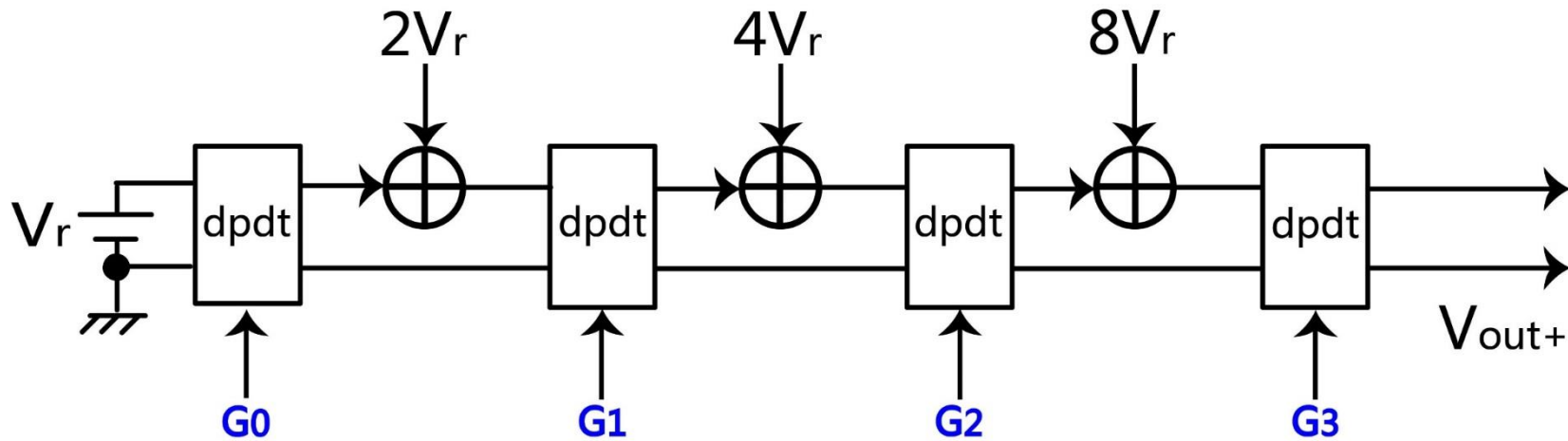


# Gray-code入力のC-DACの動作 (データ=5の場合)

eg. Data=5



### 3. Gray-code入力の電圧加算型DAC

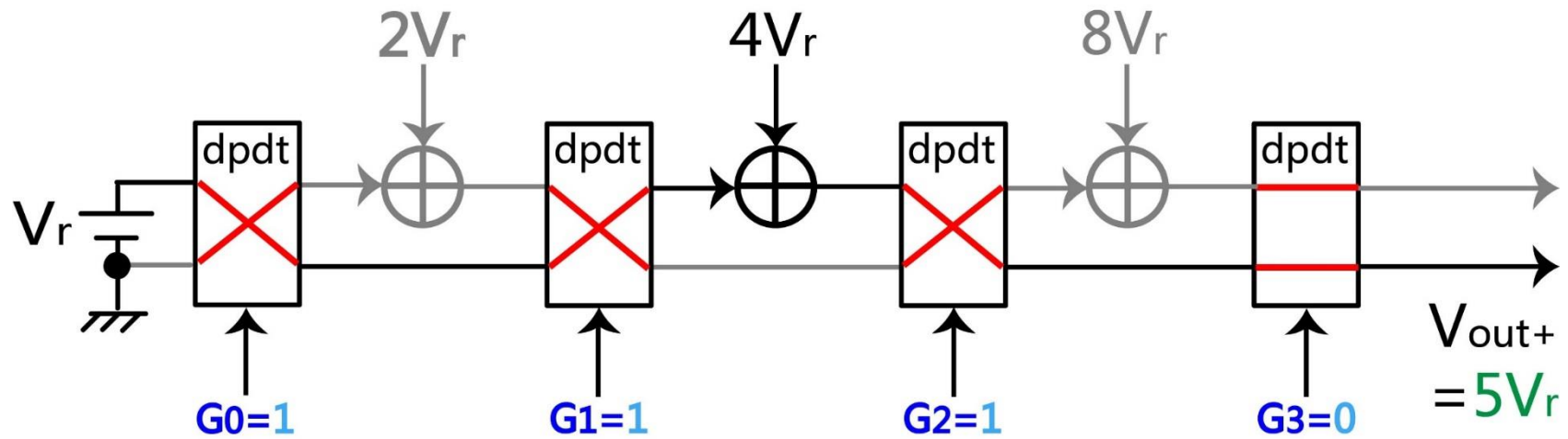


Gray-code入力の電圧加算型DAC



# Gray-code入力のV-DACの動作 (データ=5の場合)

eg. Data=5



# 目次

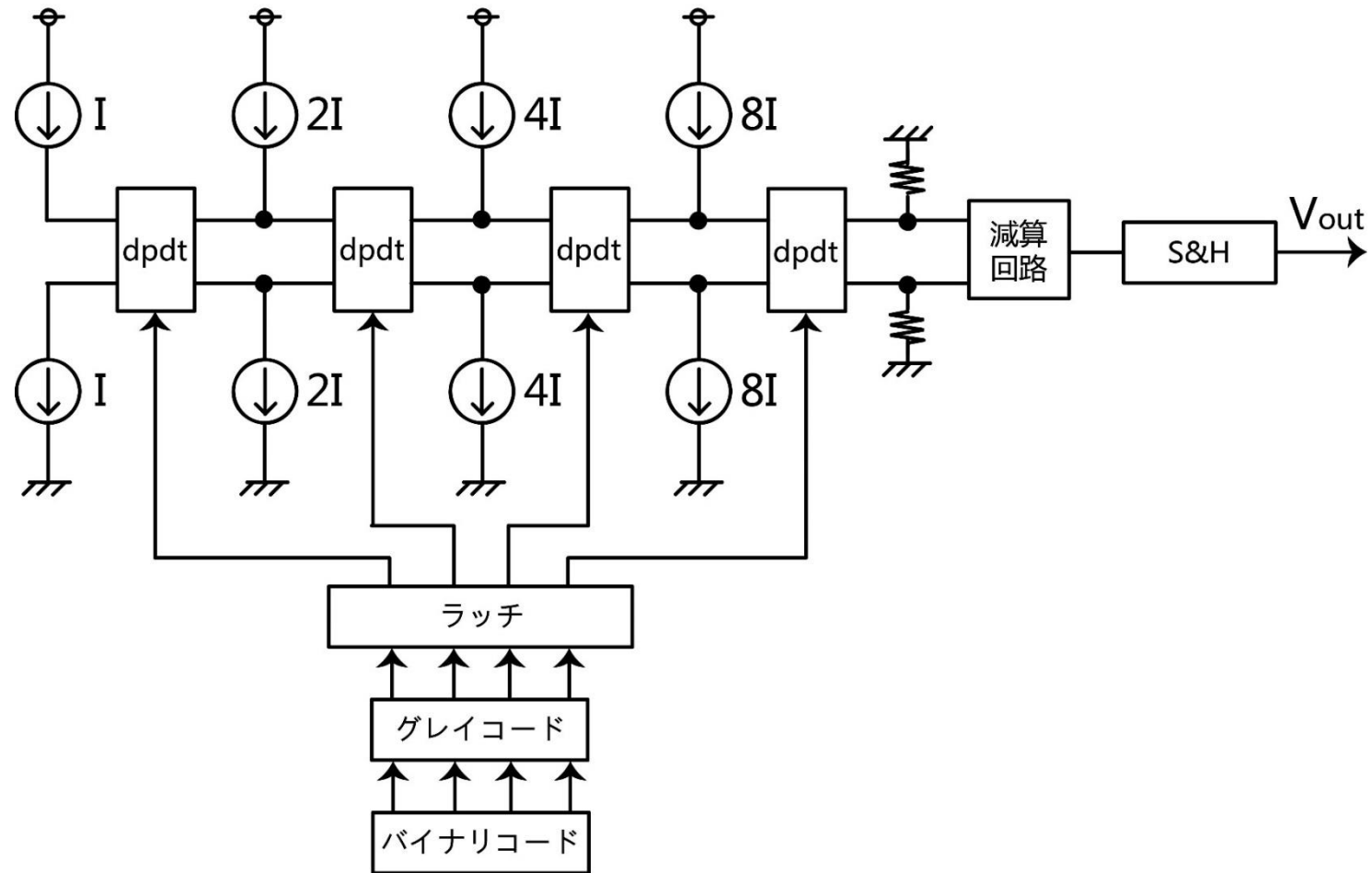
- I. 研究背景・目的
- II. 提案するGray-code入力のDACの構成と動作
- III. SPICEによるシミュレーション検証
- IV. まとめ

# SPICEによるシミュレーション検証

1. Gray-code入力の電流出力型DAC (I-DAC) のシミュレーション
2. Gray-code入力の容量型DAC (C-DAC) のシミュレーション
3. Gray-code入力の電圧加算型DAC (V-DAC) のシミュレーション
4. グリッチ除去の検証

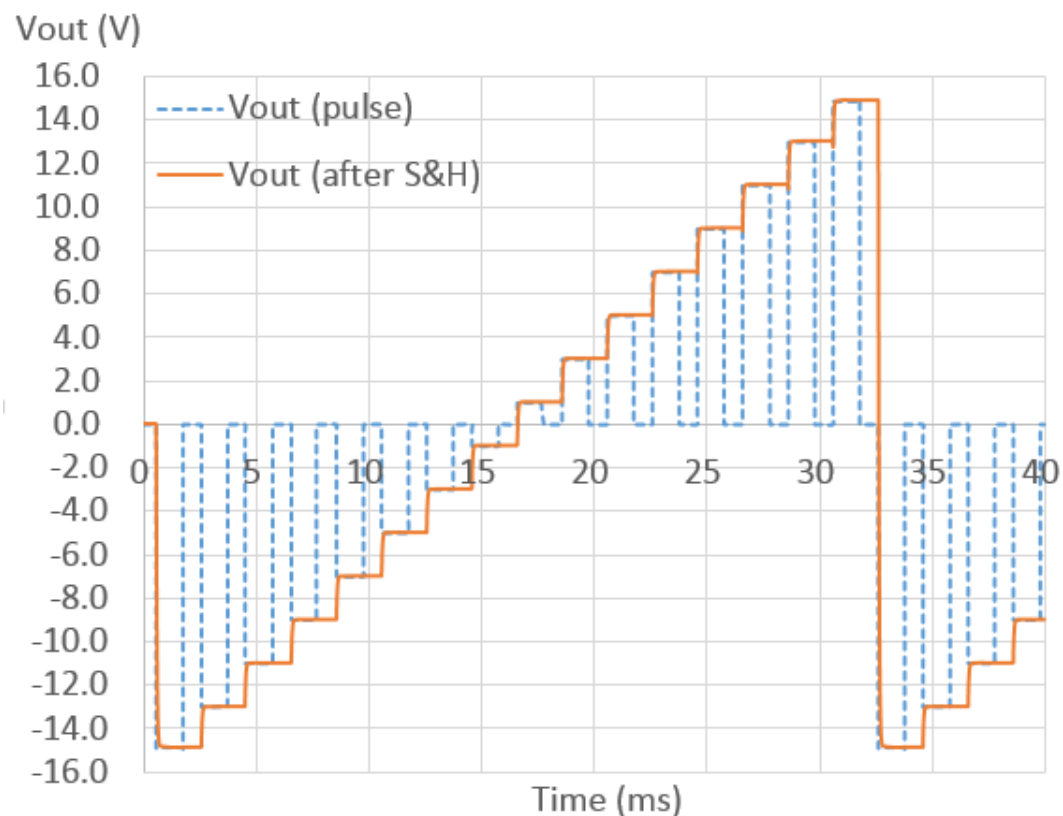
\* LTspice IVを使用

# 1. Gray-code入力のI-DACのSPICE実現

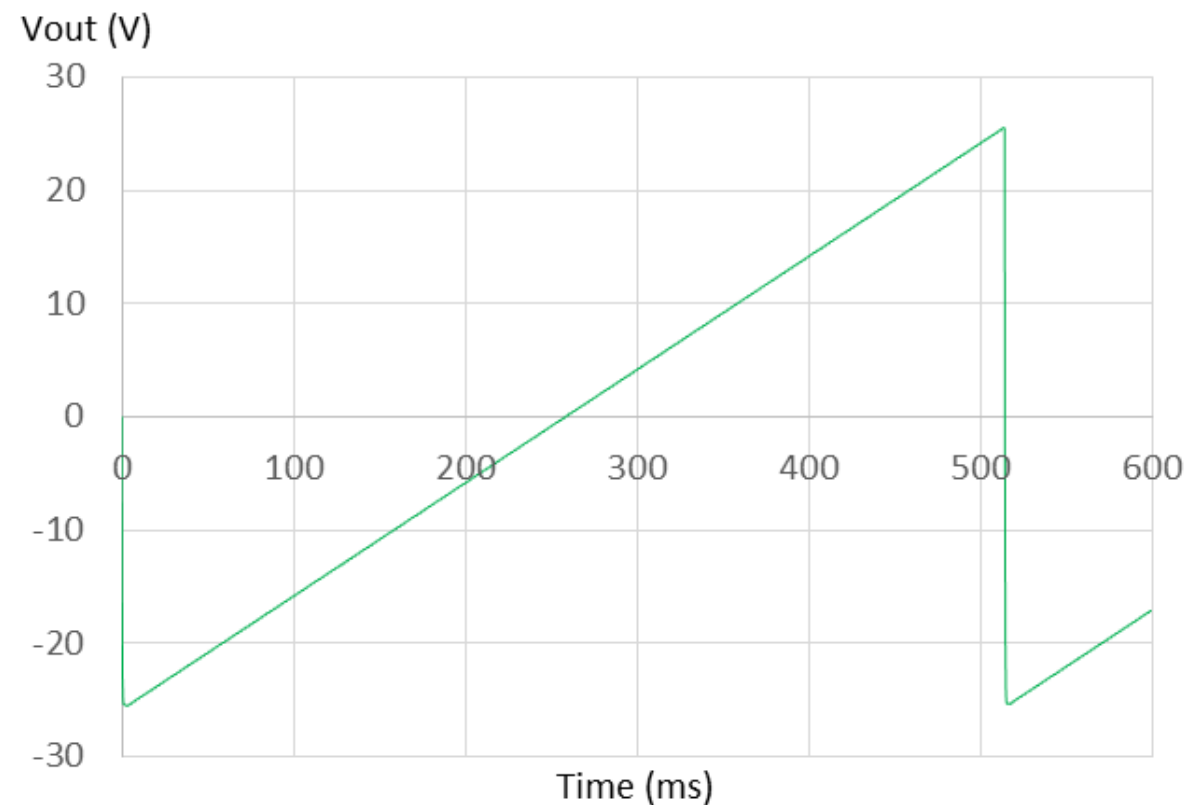


Gray-code入力の電流出力型DAC

# 1. Gray-code入力のI-DACのシミュレーション

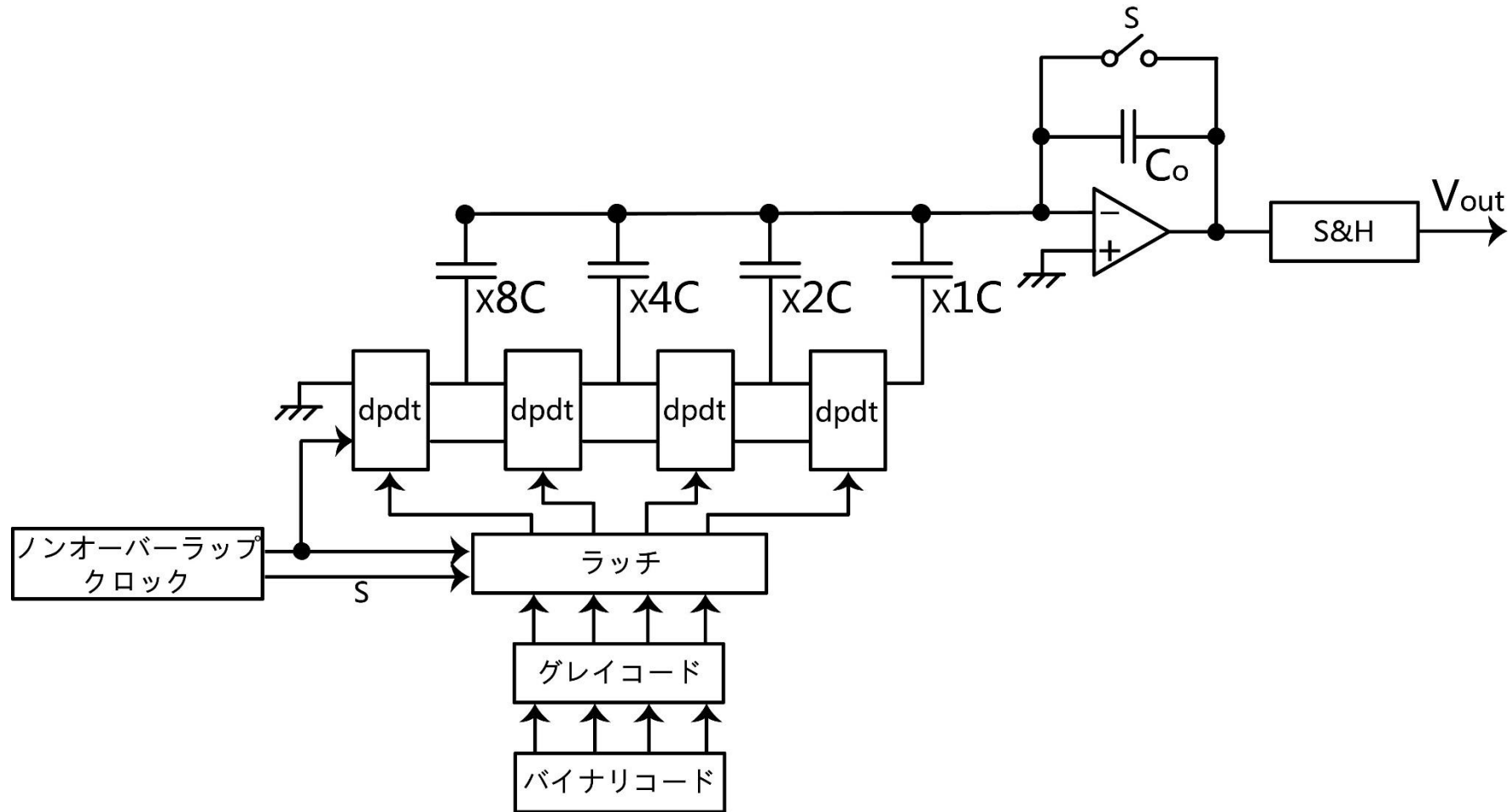


4bit I-DAC



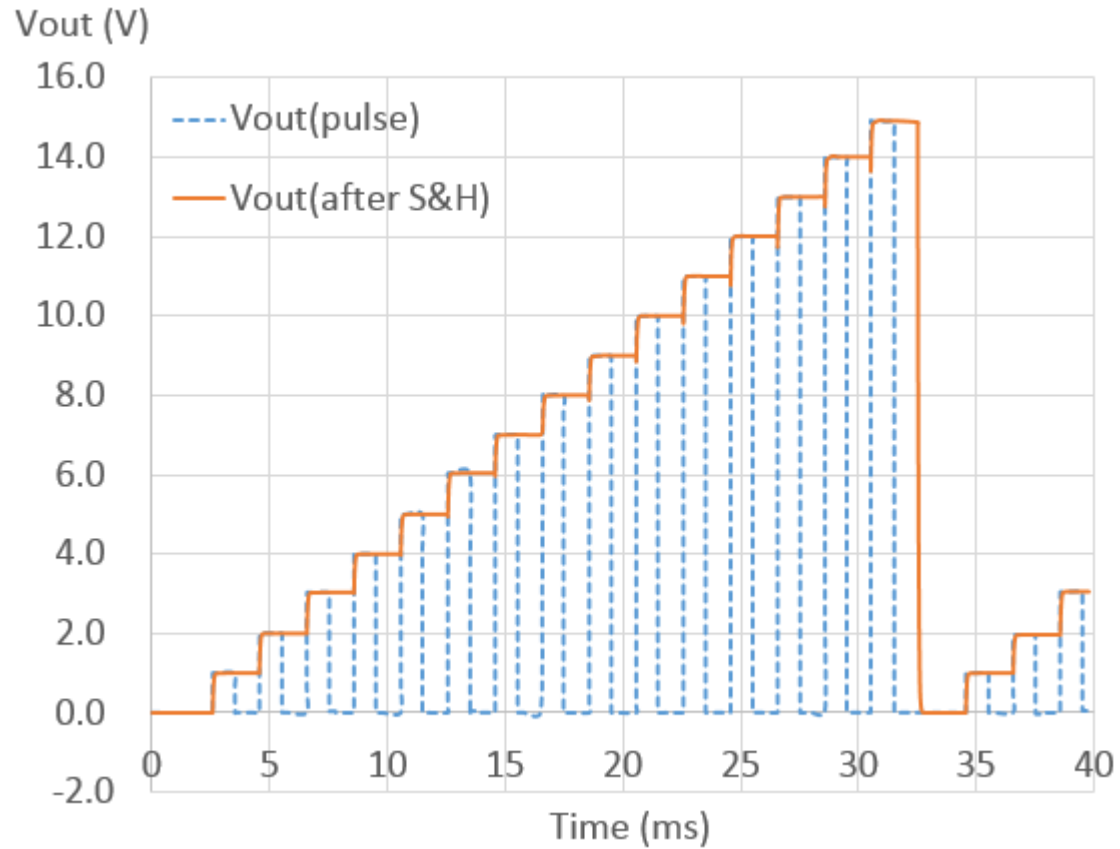
8bit I-DAC

## 2. Gray-code入力のC-DACのSPICE実現

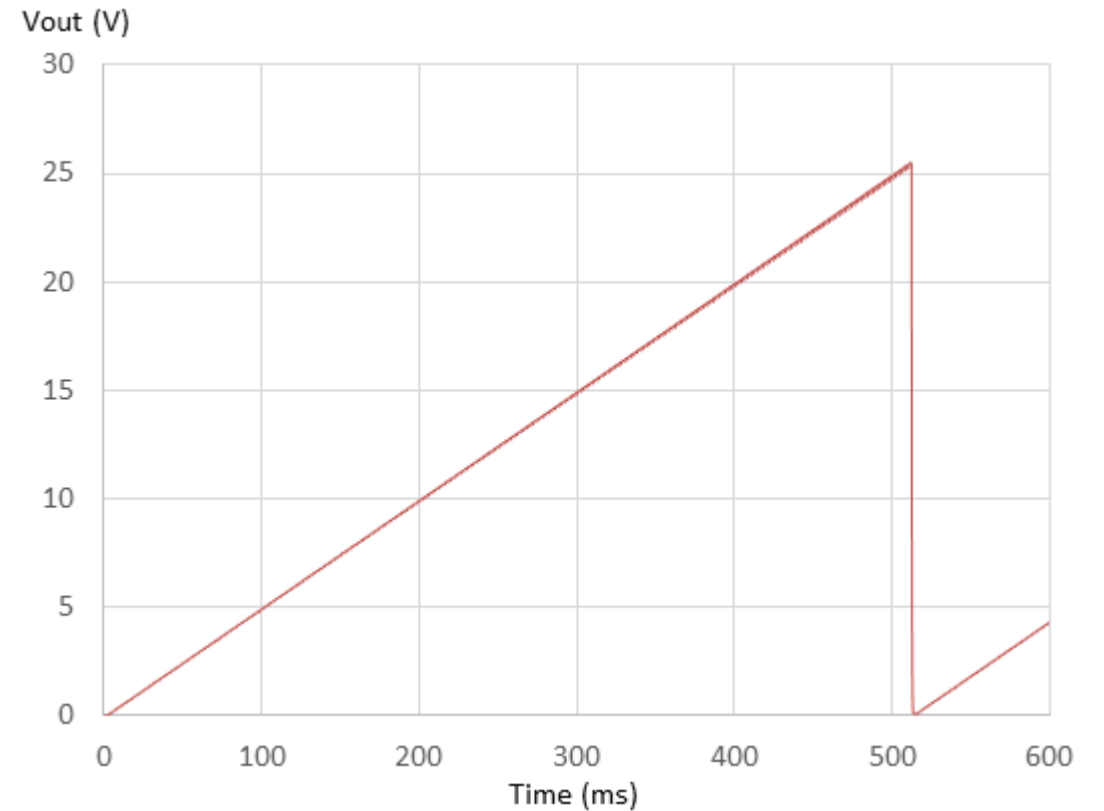


Gray-code入力の容量型DAC

## 2. Gray-code入力のC-DACのシミュレーション

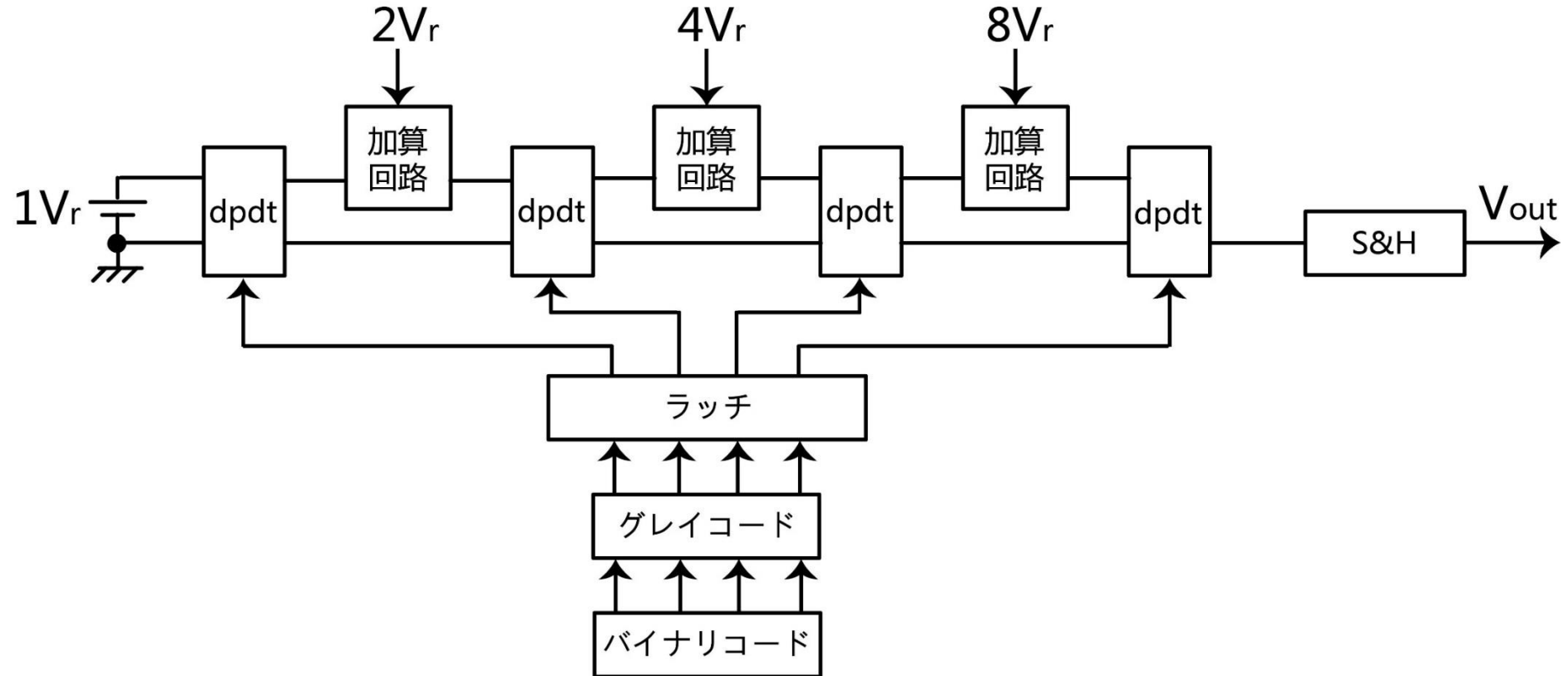


4bit C-DAC



8bit C-DAC

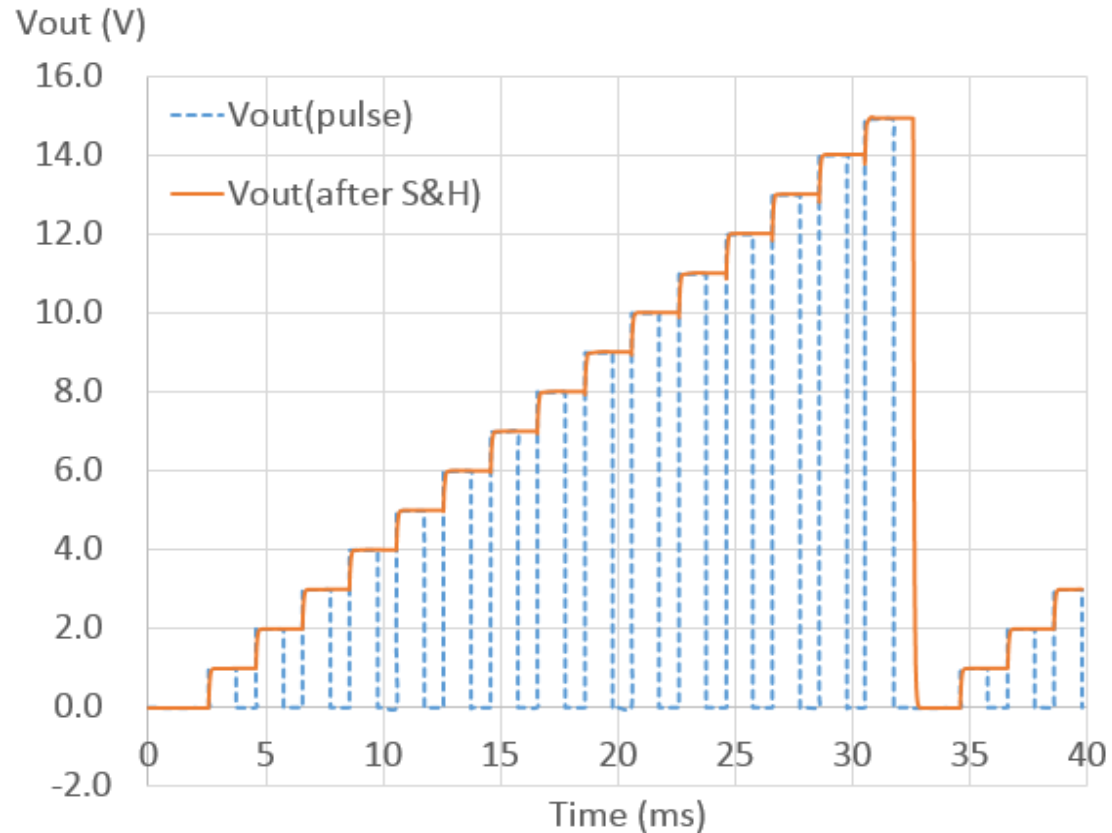
### 3. Gray-code入力のV-DACのSPICE実現



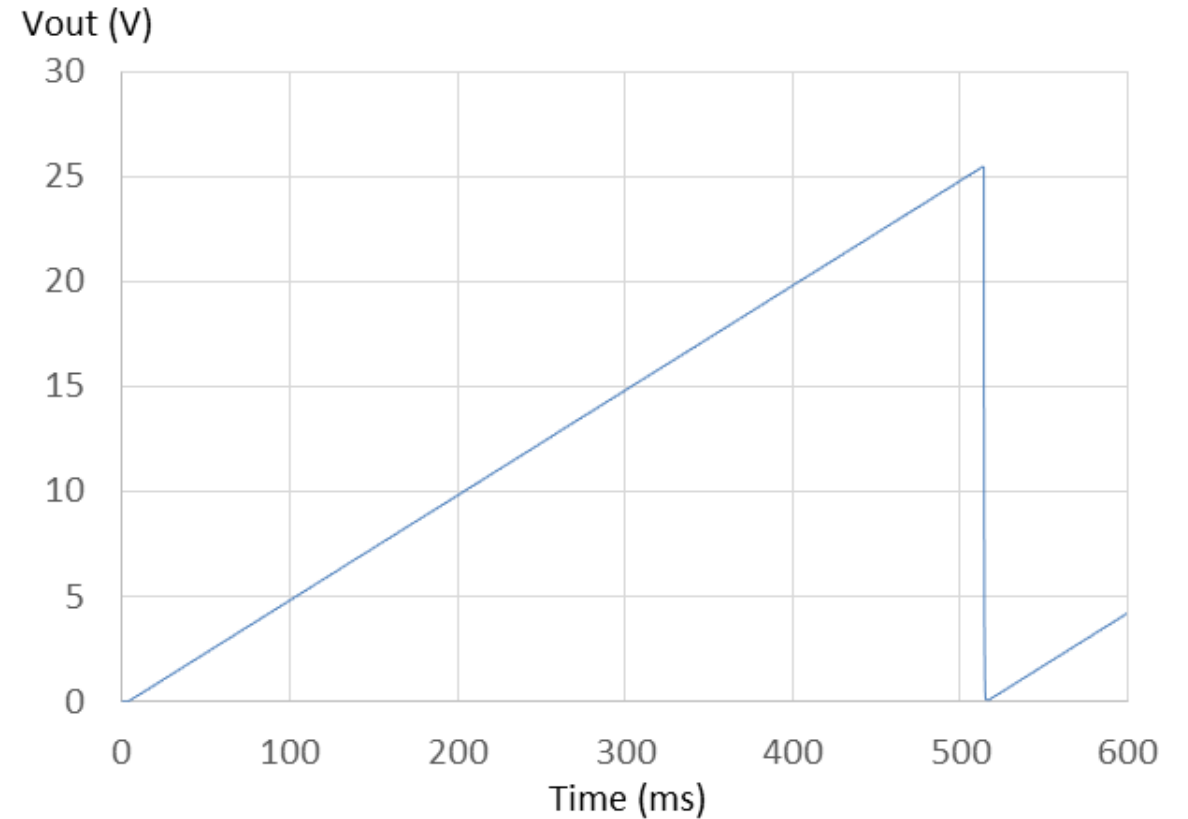
Gray-code入力の電圧加算型DAC



### 3. Gray-code入力のV-DACのシミュレーション

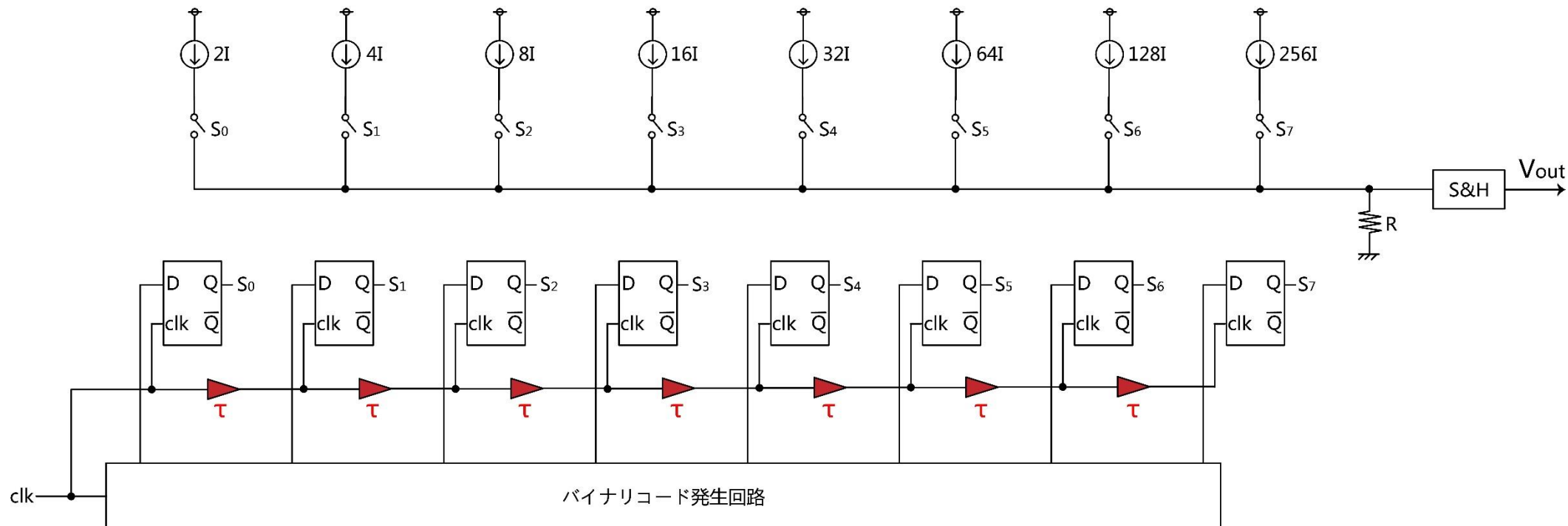


4bit V-DAC



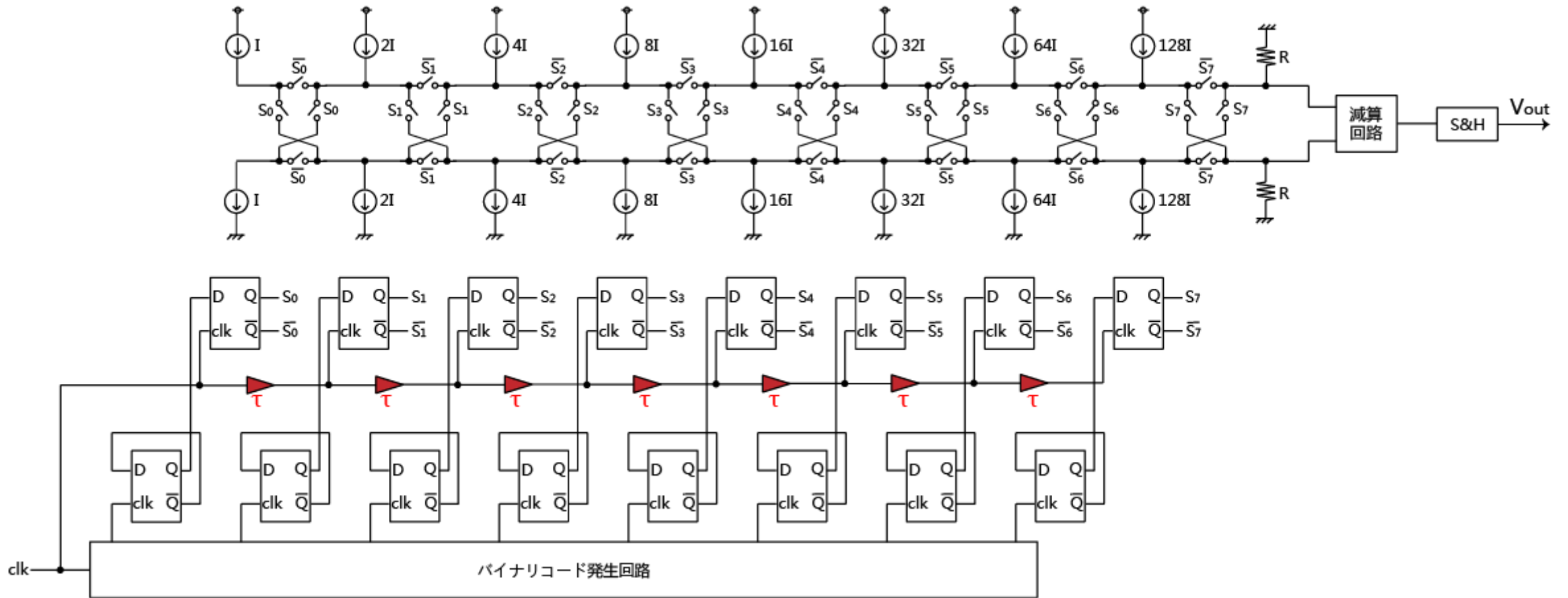
8bit V-DAC

## 4. グリッチ除去の検証



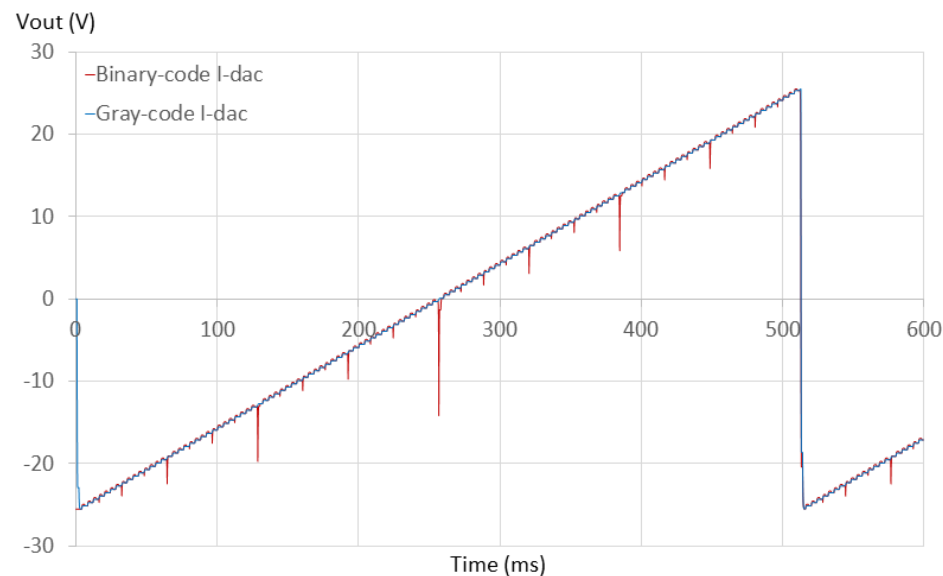
スイッチング遅延がついた従来のI-DAC (8 bit)

## 4. グリッチ除去の検証

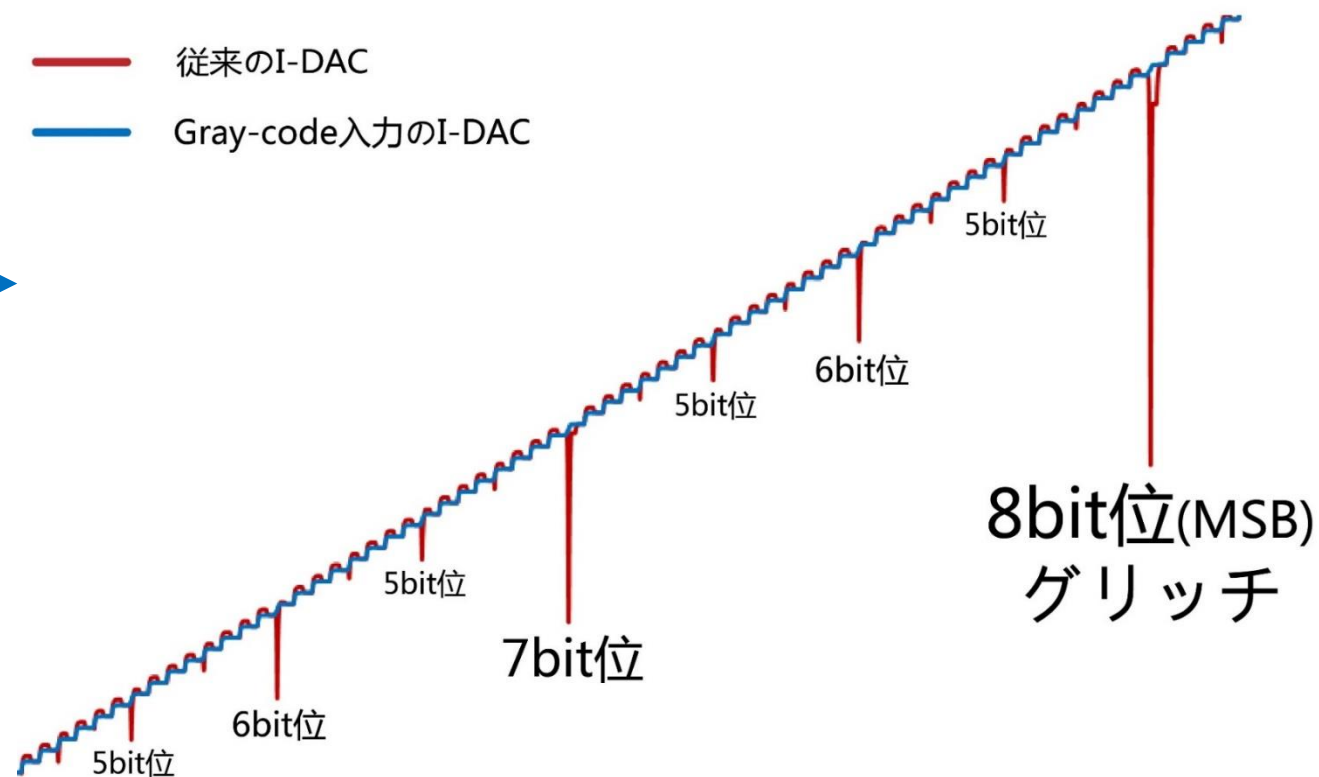


スイッチング遅延がついたGray-code入力のI-DAC (8 bit)

# アップスウィーピング

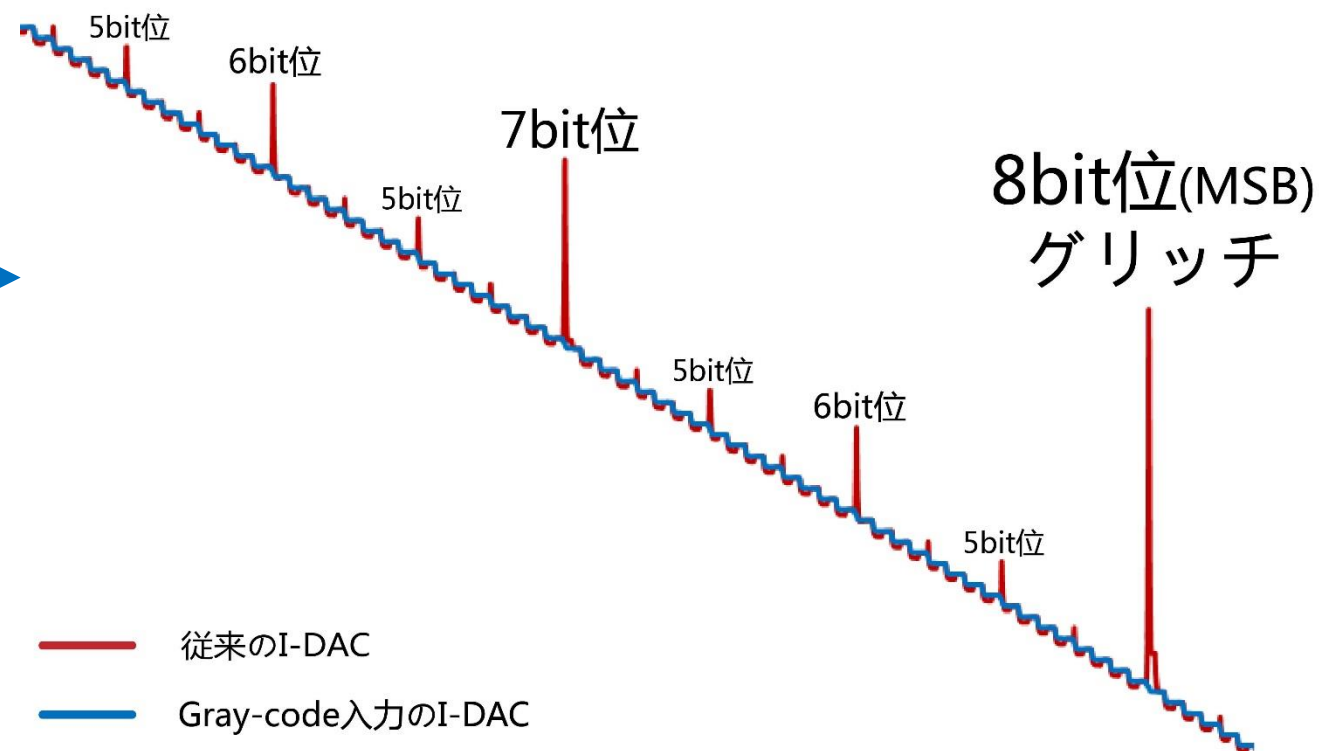
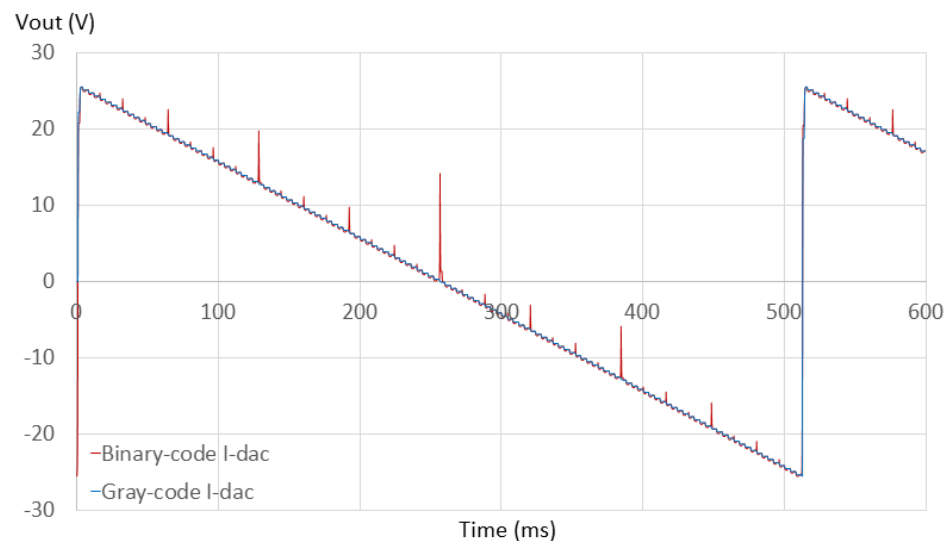


— 従来のI-DAC  
— Gray-code入力のI-DAC



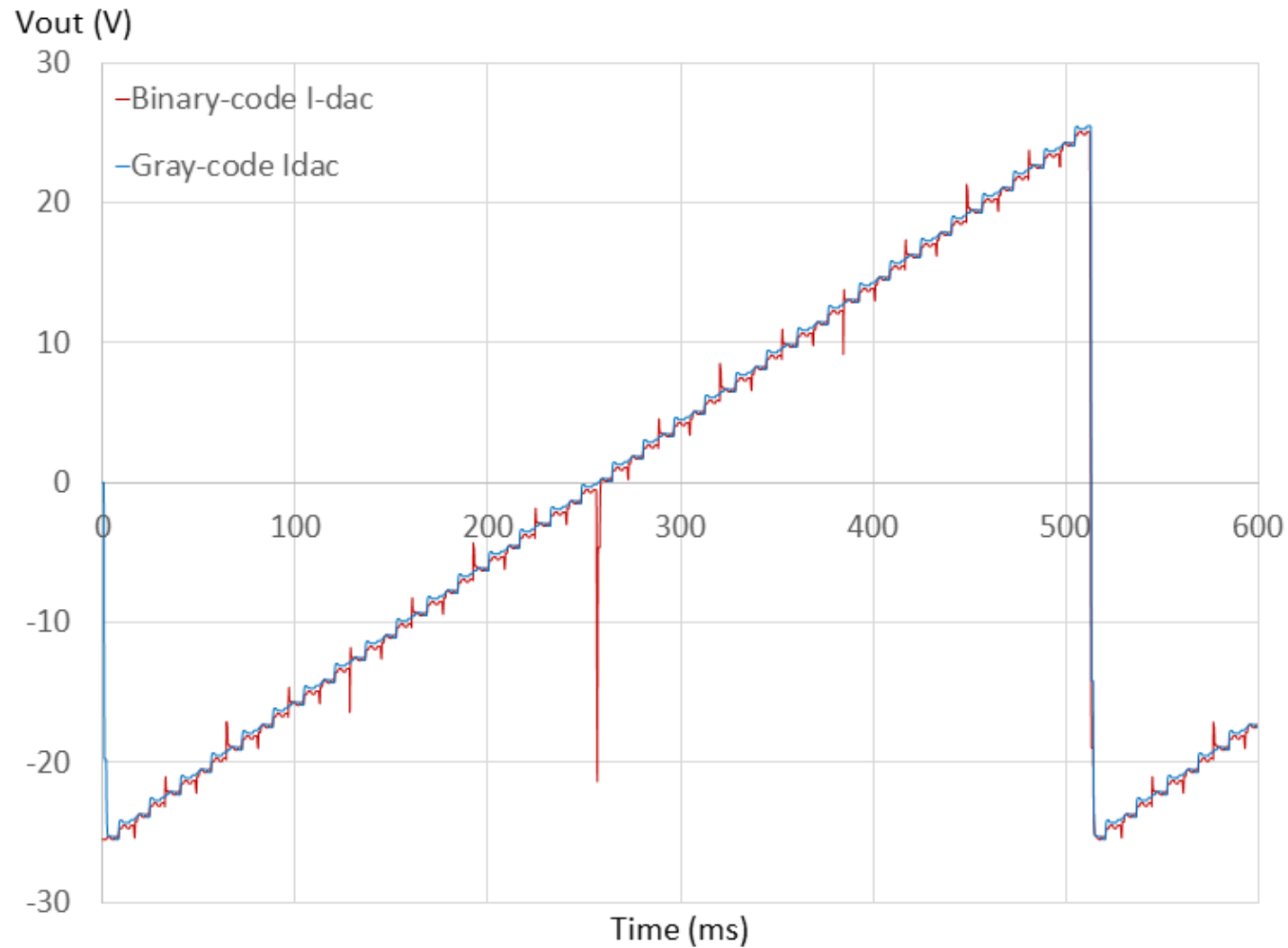
従来のI-DAC vs. Gray-code入力のI-DAC

# ダウンスウィーピング



従来のI-DAC vs. Gray-code入力のI-DAC

# ランダムなスイッチング遅延

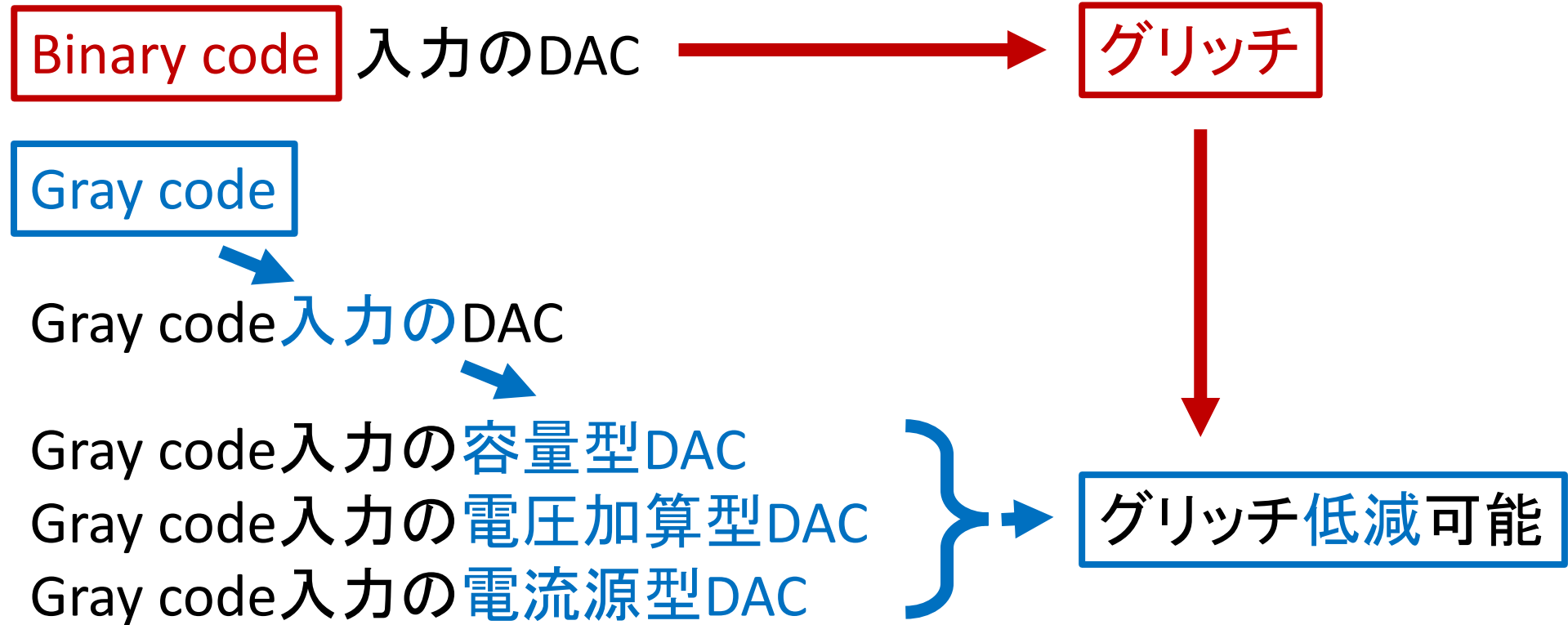


従来のI-DAC vs. Gray-code入力のI-DAC

# 目次

- I. 研究背景・目的
- II. 提案するGray-code入力のDACの構成と動作
- III. SPICEによるシミュレーション検証
- IV. まとめ

# まとめ



今後の課題:

電圧/電流スイッチマトリックスのMOSFETでの設計



## Q/A

Q: この研究は先行研究に比べて発展された点は何ですか。

A: 研究背景でお話したように、Gray-codeを入力としての綺麗な構成のDACは実現が難しいと考えられています。本論文ではGray-code入力のDACが実現できることを示します。