

高性能デジタル電源用 AD変換器とDPWM回路 の研究

群馬大学電気電子工学専攻

木村 圭吾、山田佳央、 森偉文樹、小林春夫
小堀康功、清水一也、光野正志、傘昊



OUTLINE

- 研究背景と目的
- A/D変換器の低レテンシー化
- デジタルPWM(DPWM)発生回路
- DPWM設計アルゴリズム
- まとめと今後の課題



OUTLINE

- **研究背景と目的**
- A/D変換器の低レテンシー化
- デジタルPWM(DPWM)発生回路
- DPWM設計アルゴリズム
- まとめと今後の課題

研究背景

■電源回路への要求大

- 近年、LSIの超大規模化・微細化、マイクロプロセッサ性能向上
 - ・スイッチング時間高速化(数十psec)
 - ・電源電圧の動作マージン減少(LSIの低電力化)
 - ・電圧変動量の増加(半導体微細化)

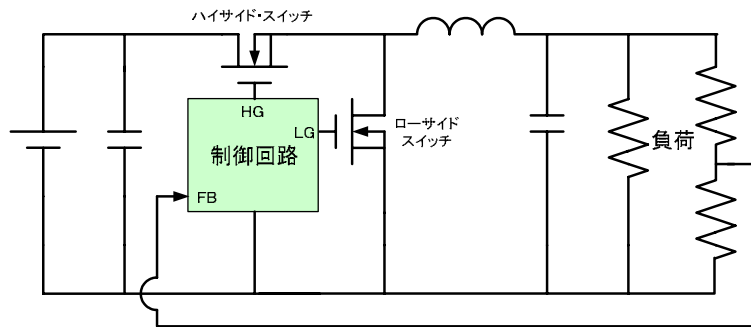
電源への要求:大

■デジタル制御電源に関心

- 電源とデジタル回路の1チップシステムLSI化
- 高度な制御理論や新トポロジーの導入

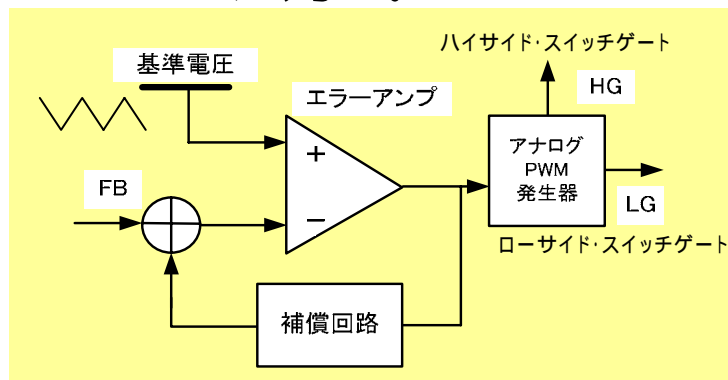
デジタル制御電源とは

■ スイッチング電源回路図

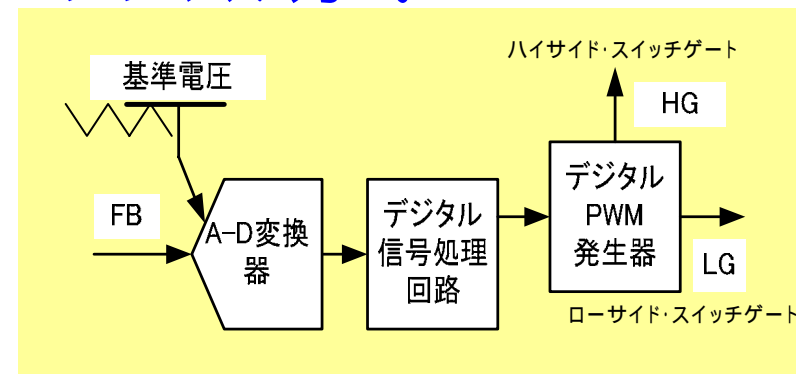


■ 制御回路部

■ アナログ方式



■ デジタル方式



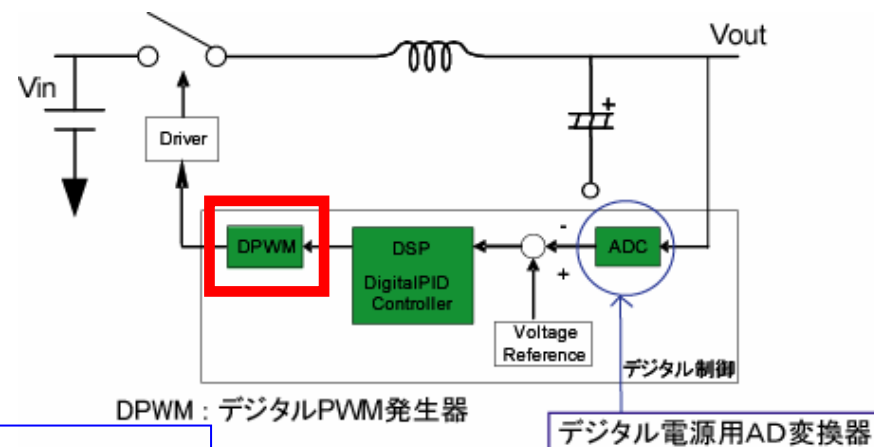
デジタル制御電源の特徴

■ メリット

- ・ 効率・応答・ノイズ性能向上
- ・ 回路変更・追加のカスタム対応力の向上
- ・ 設計スピードの向上
- ・ 小型化・低コスト化・信頼性の向上

■ デメリット

- ・ 消費電力: 大
- ・ 高コスト



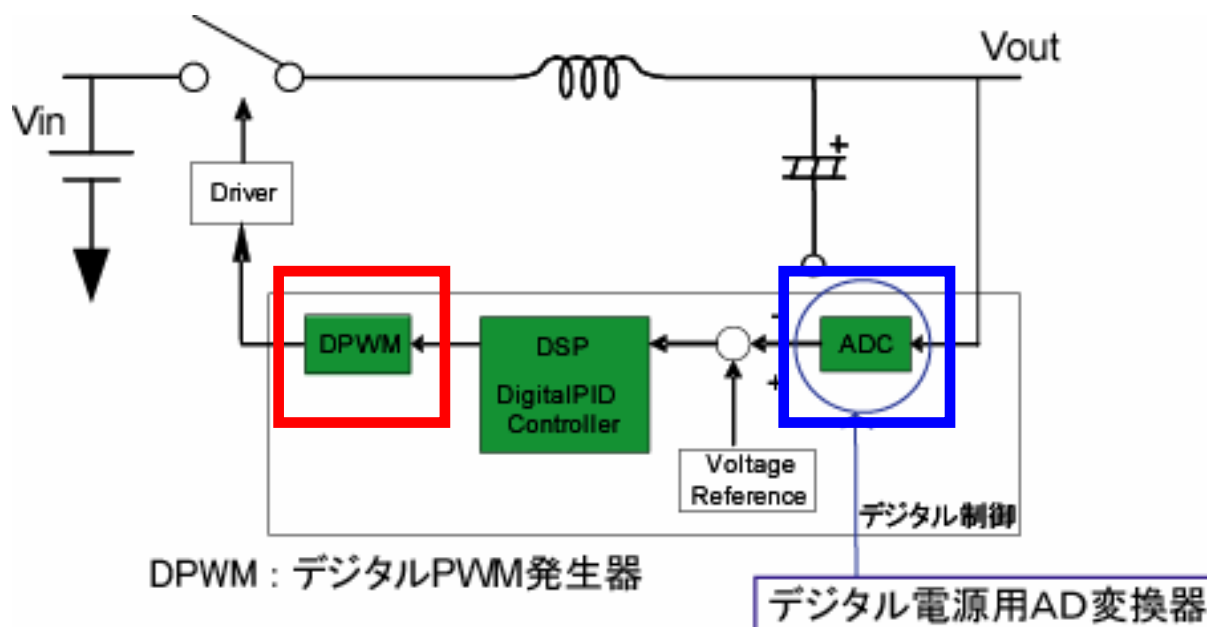
アナログとデジタルのインターフェース回路
がキーコンポーネント

研究目的

デジタル電源用キーコンポーネントの新提案

AD変換器の低レーテンシー化

高分解能DPWM回路の新アーキテクチャ

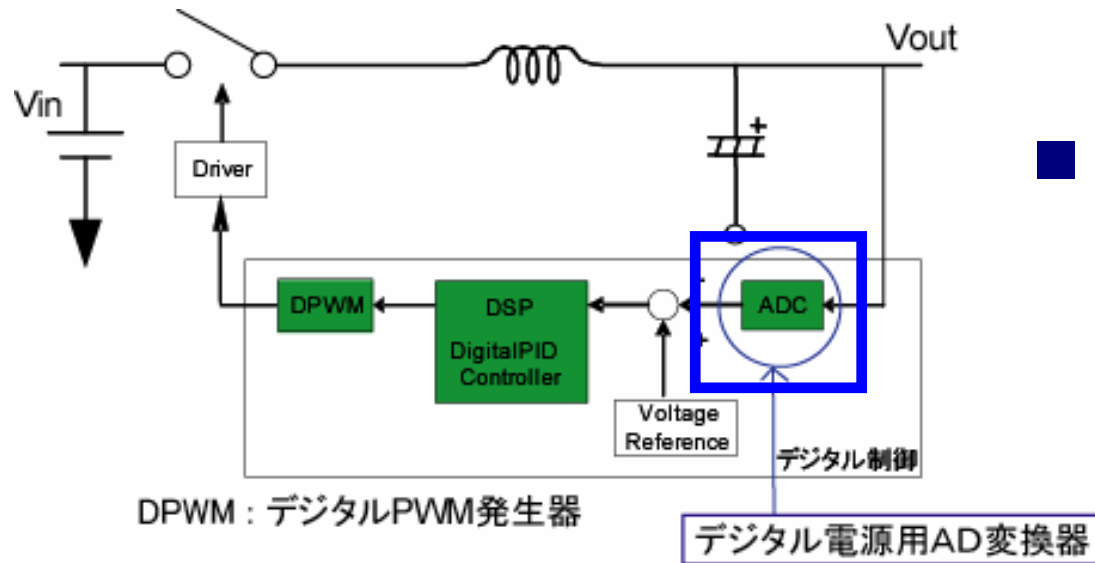




OUTLINE

- 研究背景と目的
- **A/D変換器の低レテンシー化**
- デジタルPWM(DPWM)発生回路
- DPWM設計アルゴリズム
- まとめと今後の課題

デジタル電源用A/D変換器の問題点



■ デジタル電源A/D C

逐次比較近似型
パイプライン型

- AD変換データはMSBから順次出力される。
- MSBからLSBの全てが出力されてからデジタル処理開始

← レーテンシー (latency, 遅延) が
フィードバックループの応答特性・安定性を劣化

デジタル電源用 A/D 変換器の問題点

提案

上位ビットからのビットシリアル演算
分散型積和演算
デジタルPID演算等の制御アルゴリズム

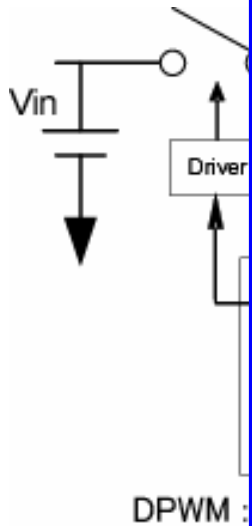
MSBの出力直後に
デジタル演算開始可能。
レーテンシーの問題軽減。

レーテンシー (latency, 遅延) が
フィードバックループの応答特性・安定性を劣化

- AD変換
- MSB

原ADC
近似型
ン型

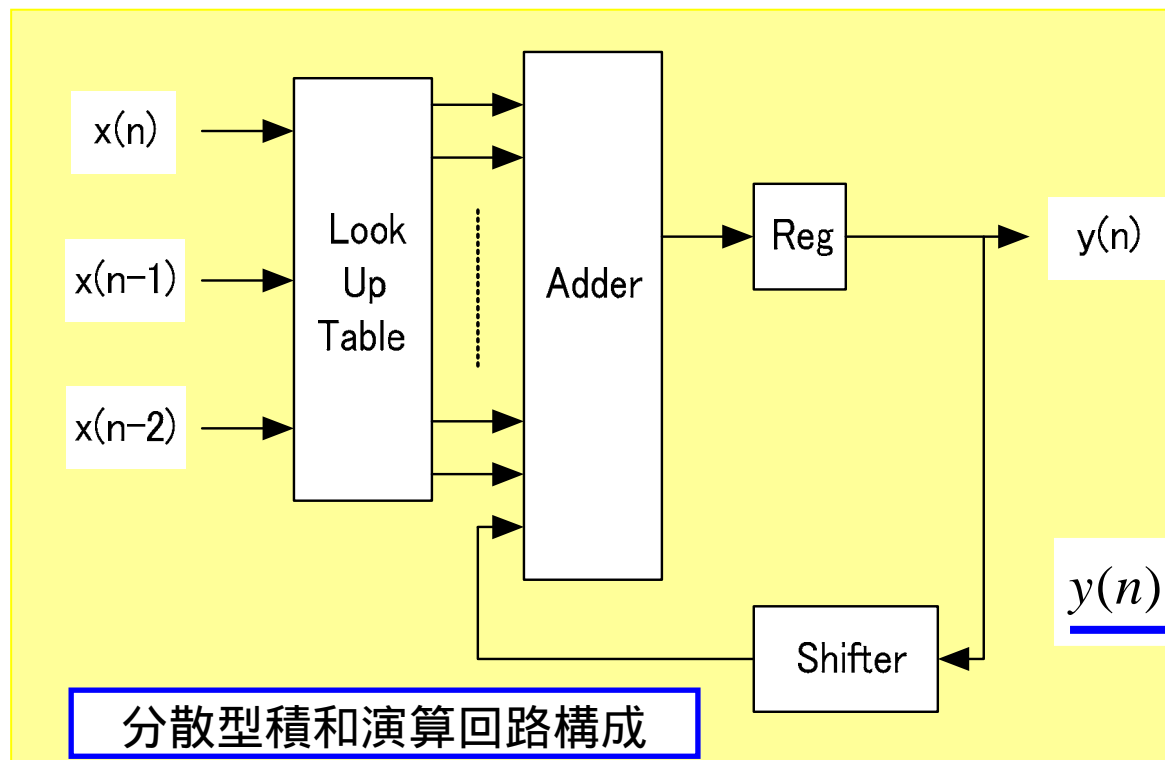
理開始



デジタル電源用 A/D 変換器の低レーテンシー化

分散型積和演算

定係数の積和演算をルックアップ・テーブル
(ROM等)により実現する計算手法



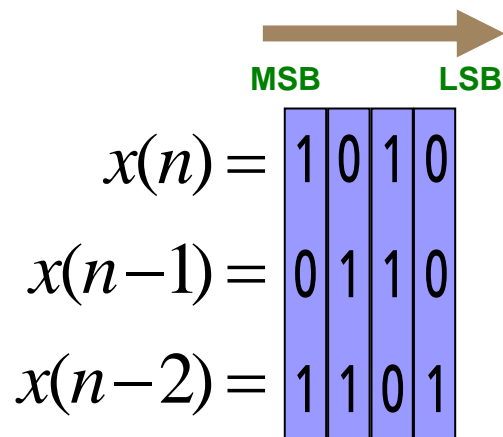
上位ビットからの
シリアル演算

$$y(n) = h_0 x(n) + h_1 x(n-1) + h_2 x(n-2)$$

デジタル入力

分散型積和演算 動作原理

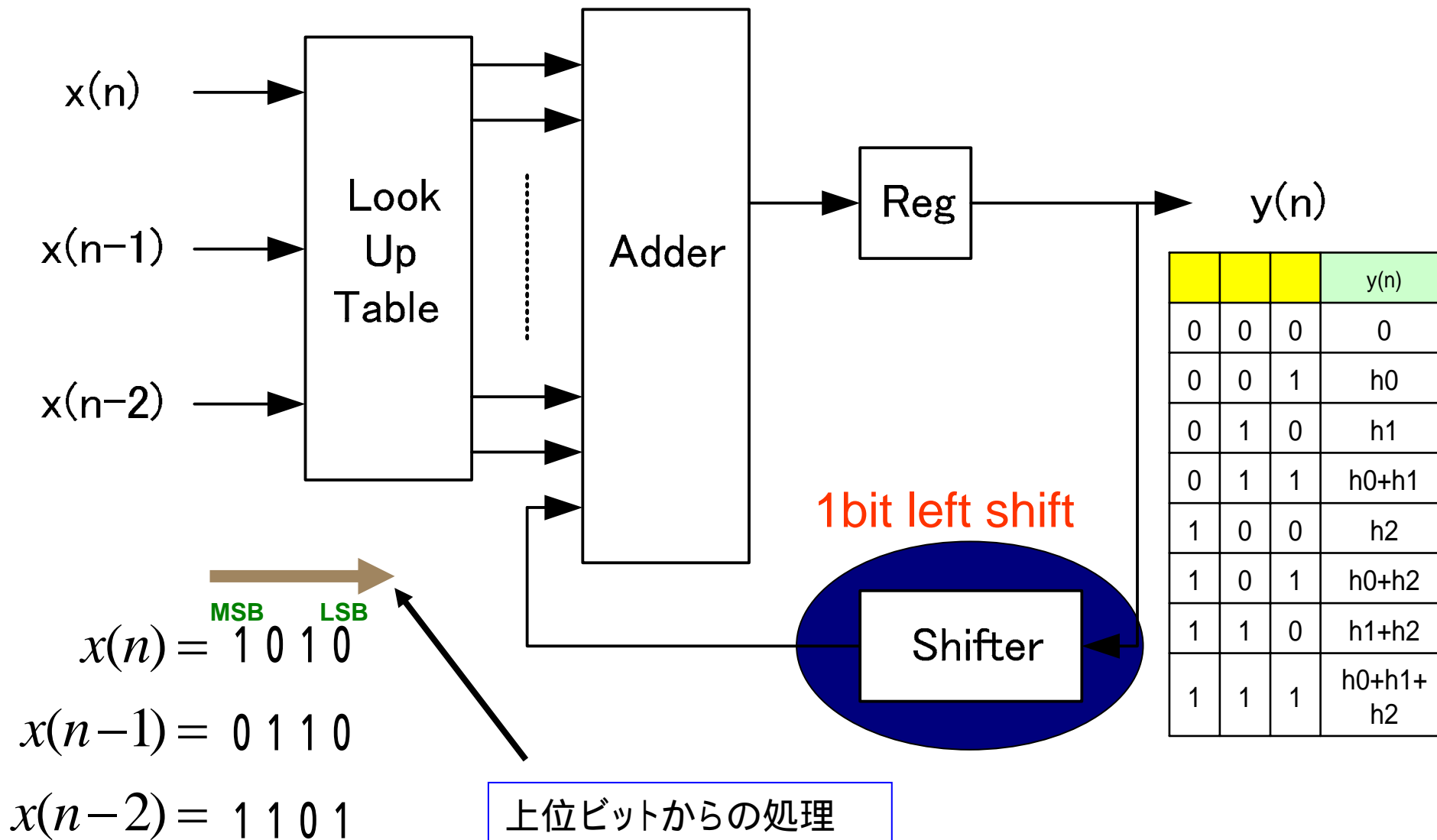
$$y(n) = \underline{h_0}x(n) + \underline{h_1}x(n-1) + \underline{h_2}x(n-2)$$



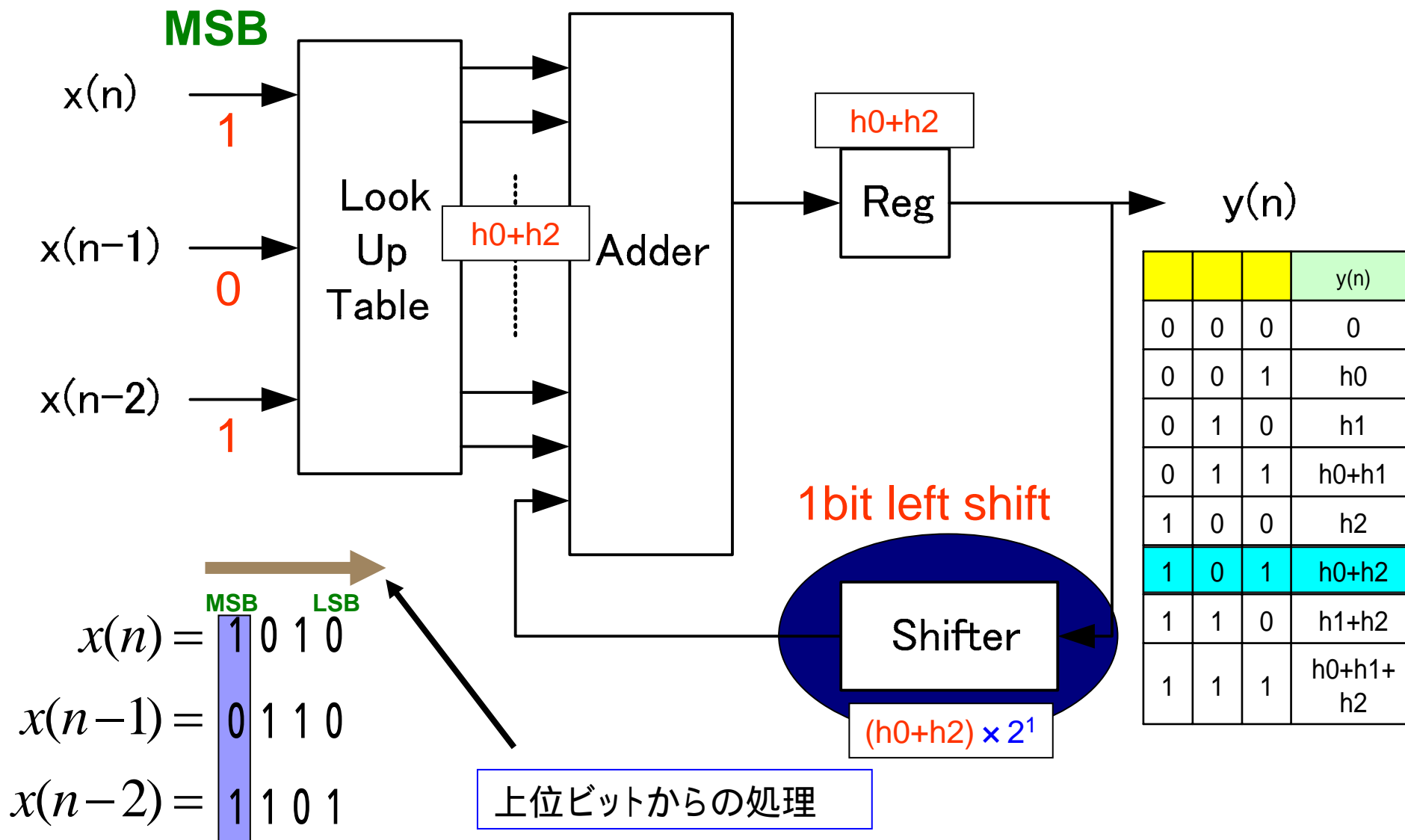
Look Up Table

			y(n)
0	0	0	0
0	0	1	h0
0	1	0	h1
0	1	1	h0+h1
1	0	0	h2
1	0	1	h0+h2
1	1	0	h1+h2
1	1	1	h0+h1+h2

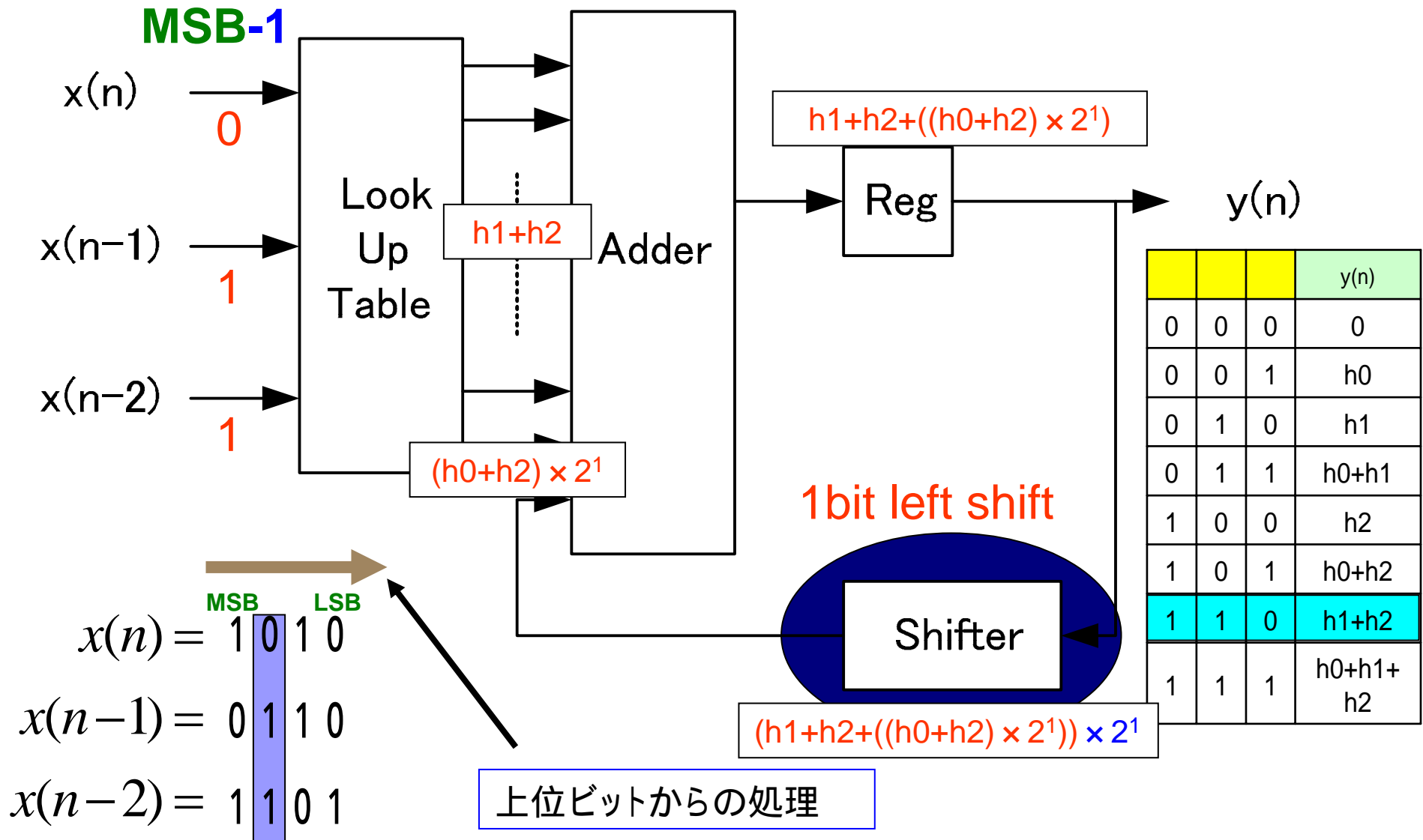
分散型積和演算 動作原理



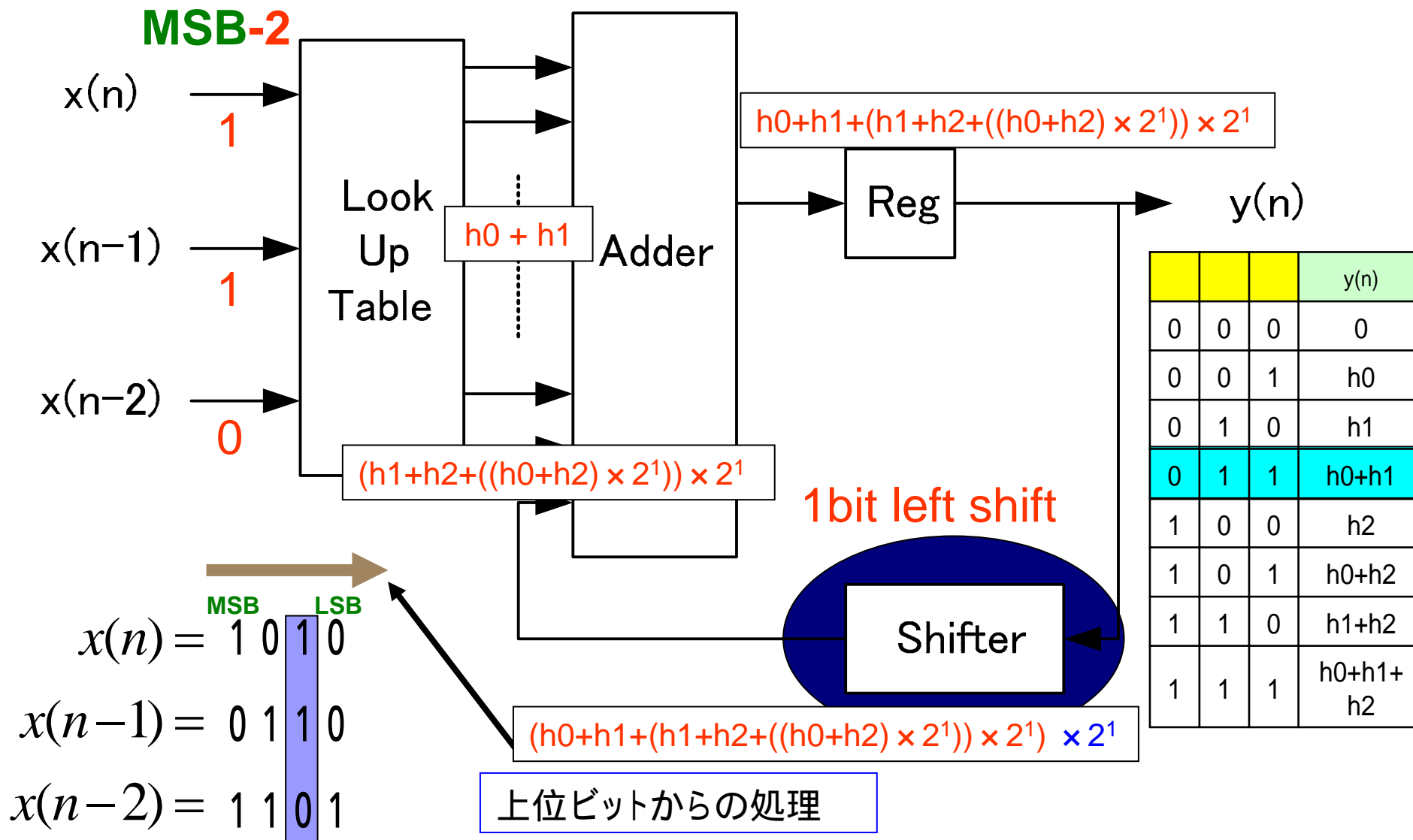
分散型積和演算 動作原理



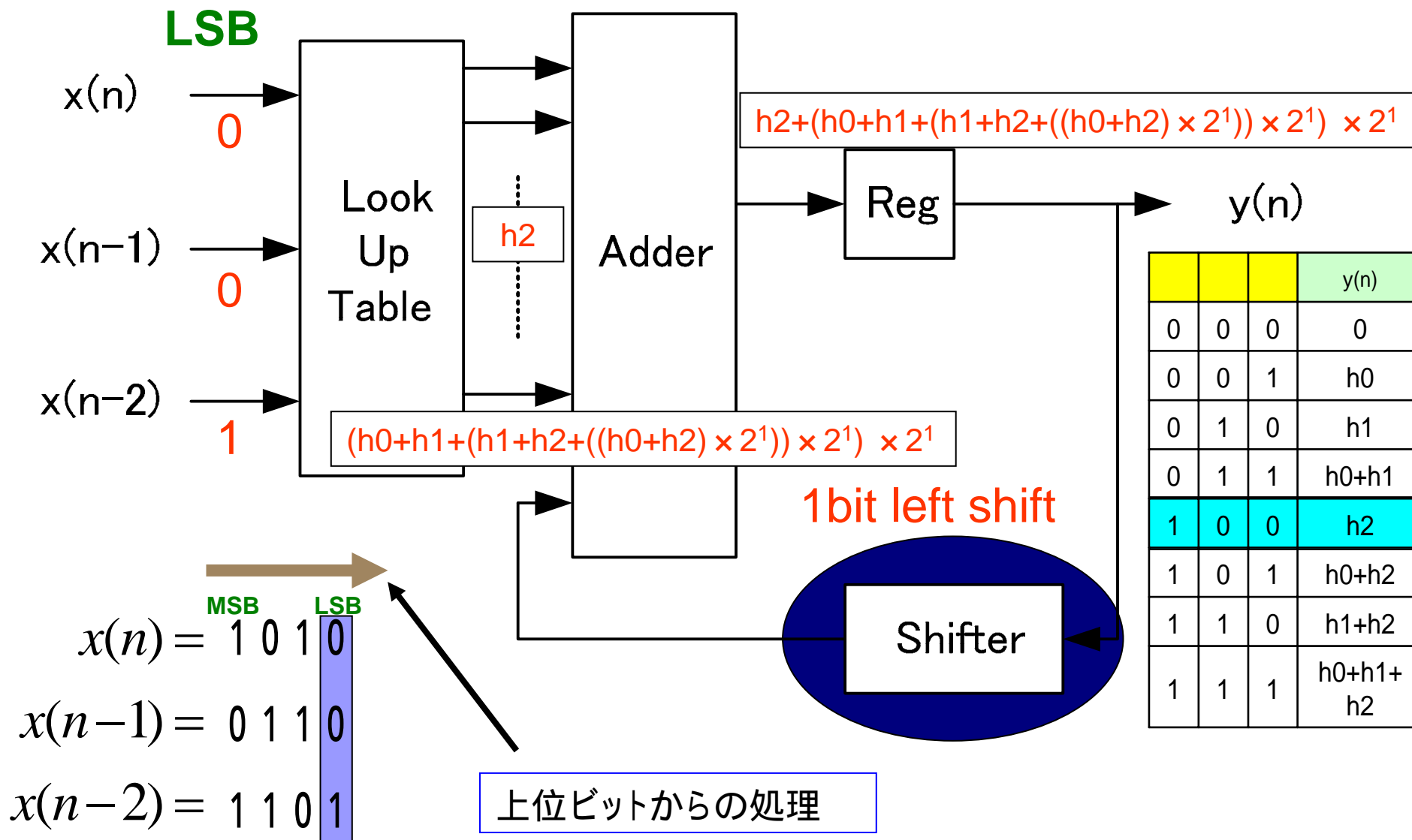
分散型積和演算 動作原理



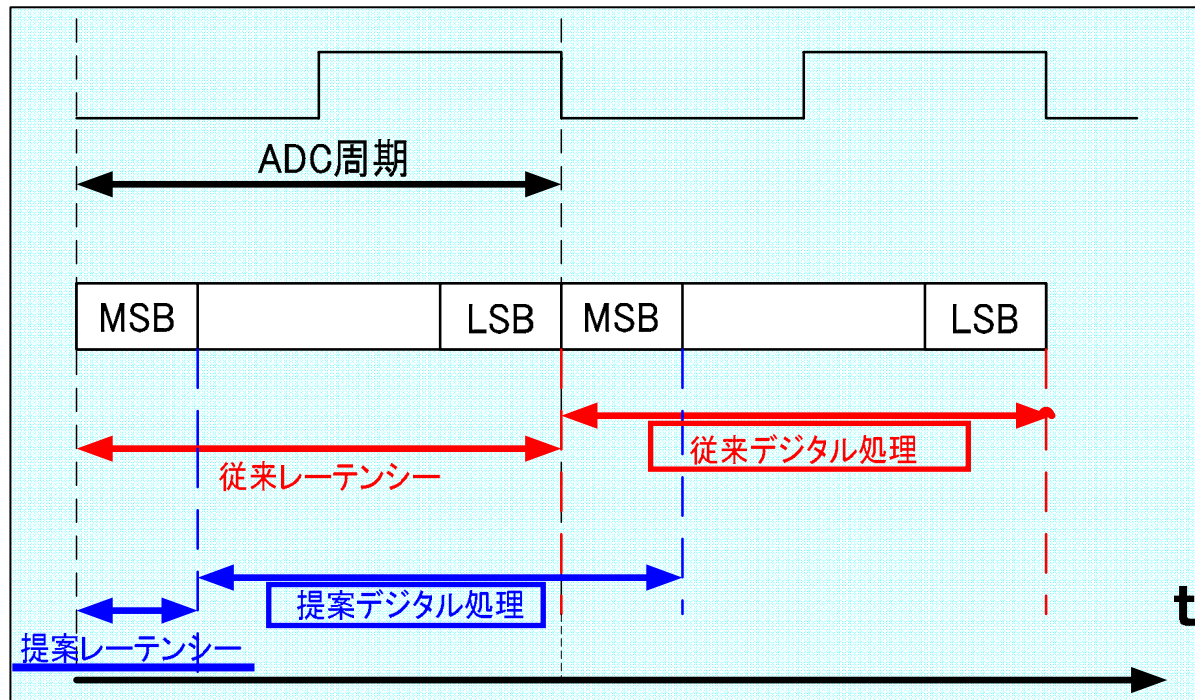
分散型積和演算 動作原理



分散型積和演算 動作原理



デジタル電源用AD変換器の低レーテンシー化 ～ タイミングチャート～



例) 12bit分解能のとき

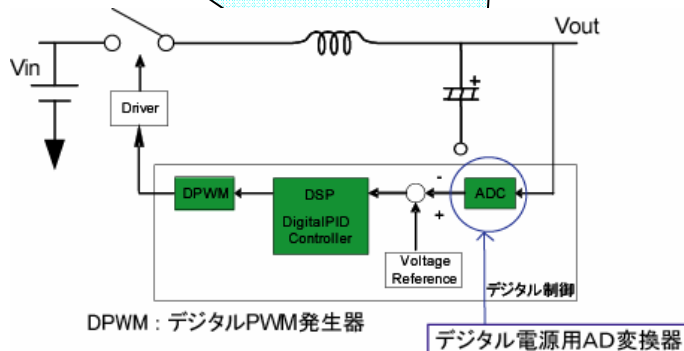
デジタル演算開始

従来 : 12bit全て出力後

提案 : 最初のビット (MSB) 出力後

効果

・フィードバックループでの
レーテンシー(遅れ)が小





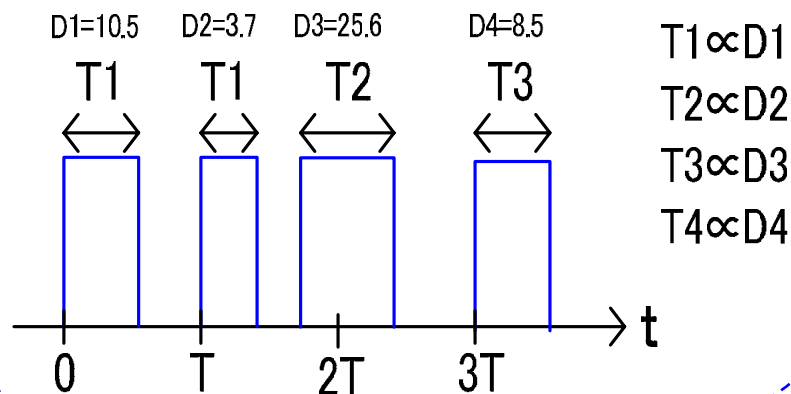
OUTLINE

- 研究背景と目的
- AD変換器の低レーテンシー化
- **デジタルPWM (DPWM) 発生回路
目標仕様及び回路構成**
CoarseDTC
FineDTC
設計結果
- DPWM設計アルゴリズム
- まとめと今後の課題

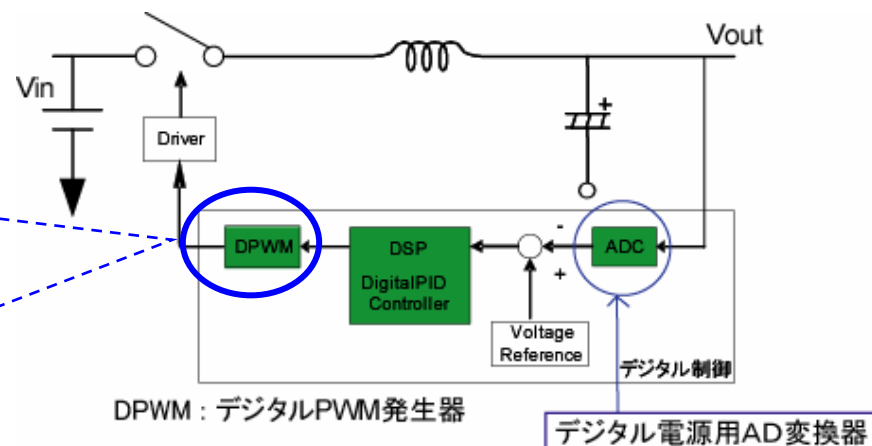
D PWMと時間分解能

■デジタルPWM (DPWM)

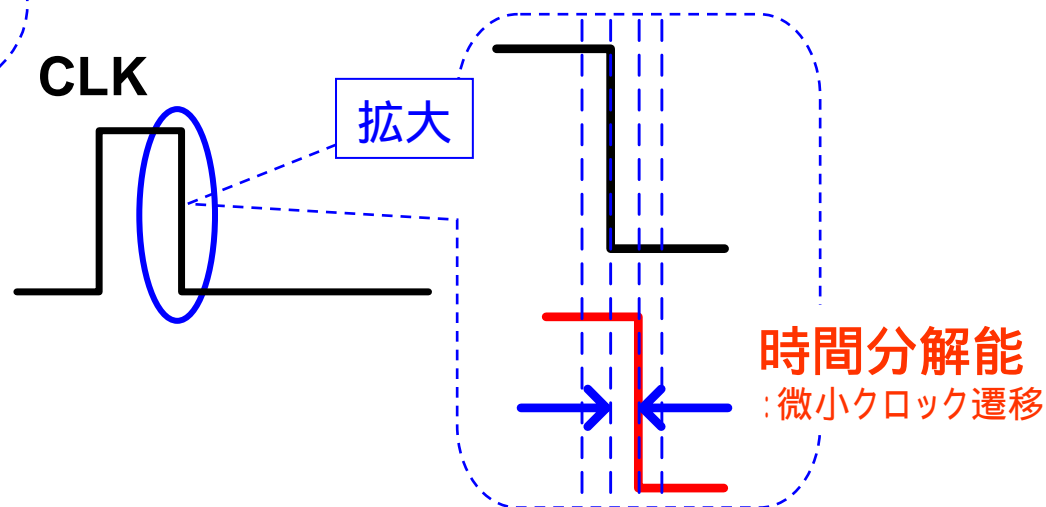
デジタル入力 時間出力:変換回路



デジタル入力と
PWMデューティ比は比例関係.



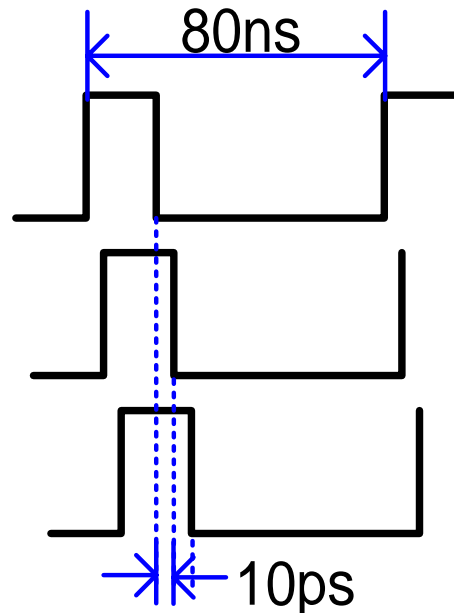
■PWM・・・パルス幅変調
(振幅からスイッチのON時間の長さで波形を生成)



DPWM全体の目標仕様

■ 目標仕様

- 1周期80ns
(ADCサンプリング周波数=12.5MHz)
- 使用可能クロック: 100MHz, 12.5MHz
- 1LSB(時間分解能): 10ps
- DPWM分解能: 13bit



・デジタルPWM: 十分な制御性能・・・12bit以上

→ 分解能: 13bit

$$\frac{80ns}{2^{13}} = \frac{80ns}{8192} = 10ps$$

(従来例: 米TI社電源デジタル制御IC(2005)
時間分解能 150ps)

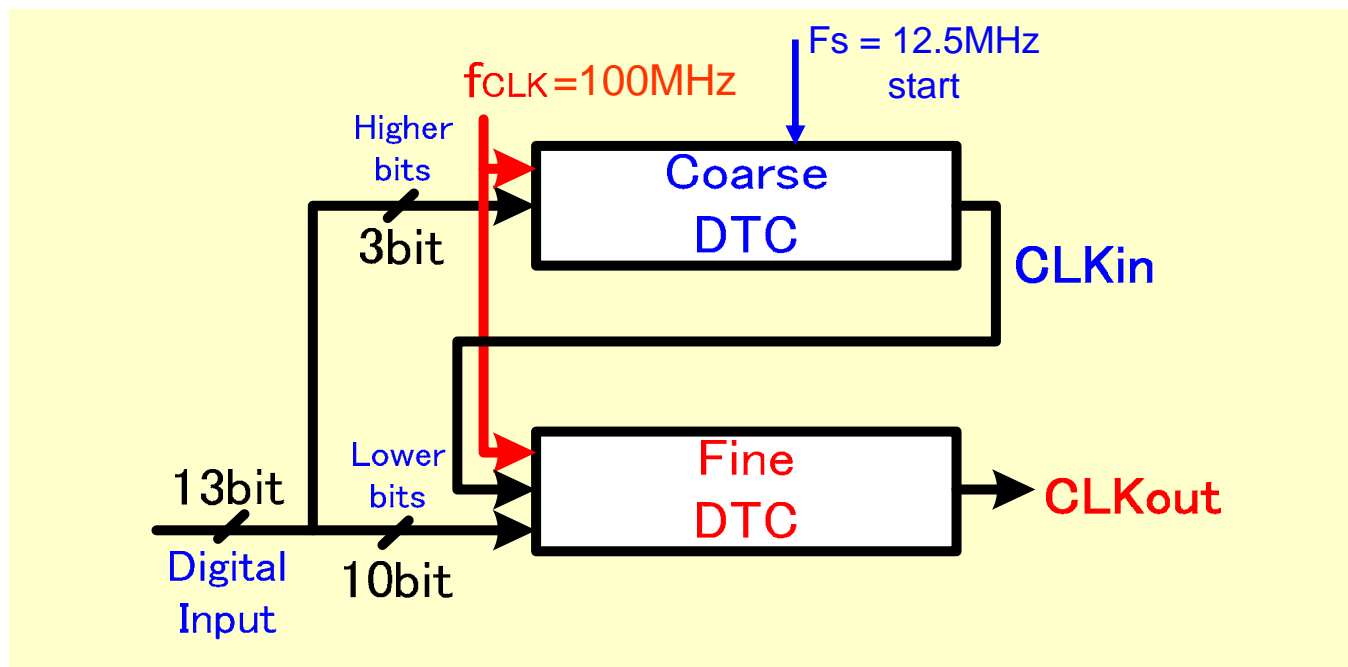
DPWM全体回路構成

■ DPWM全体回路構成

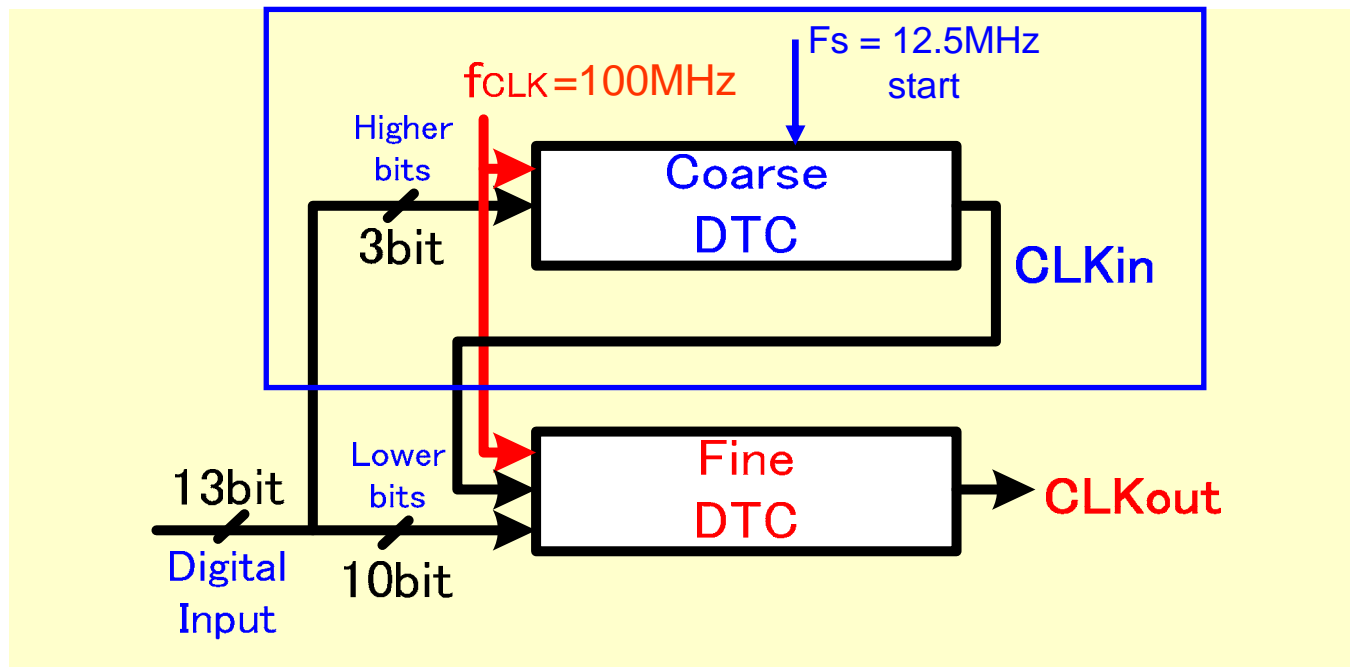
■ Digital Input: 13[bit]

- 上位3ビット・・・Coarse DTC
- 下位10ビット・・・Fine DTC

■ DTC (Digital-to-Time Converter) : デジタル入力 時間出力



■ CoarseDTCについて



CoarseDTCの仕様

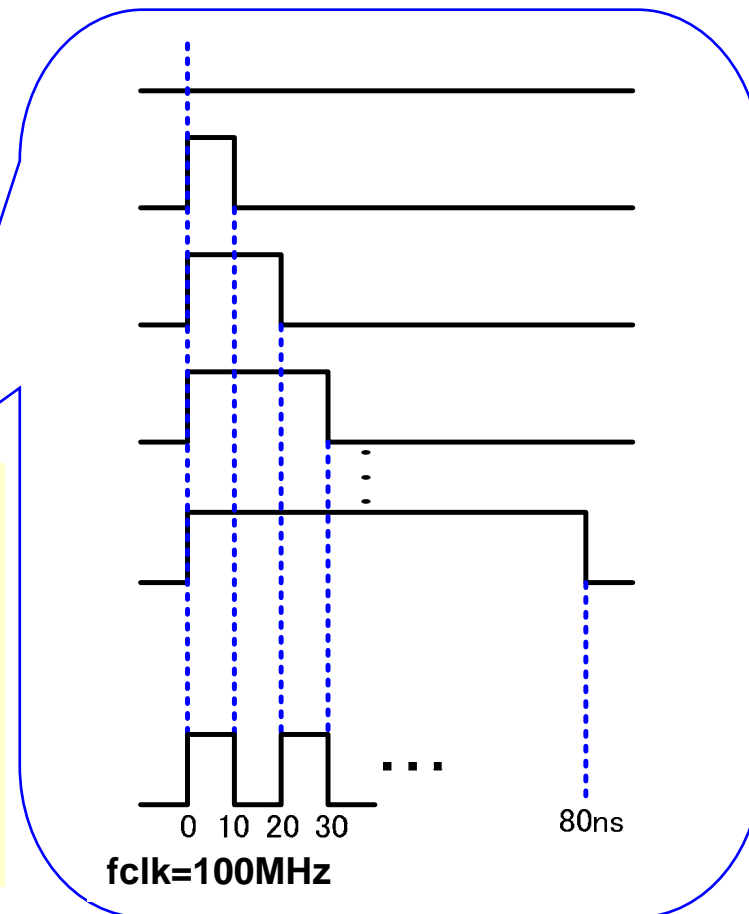
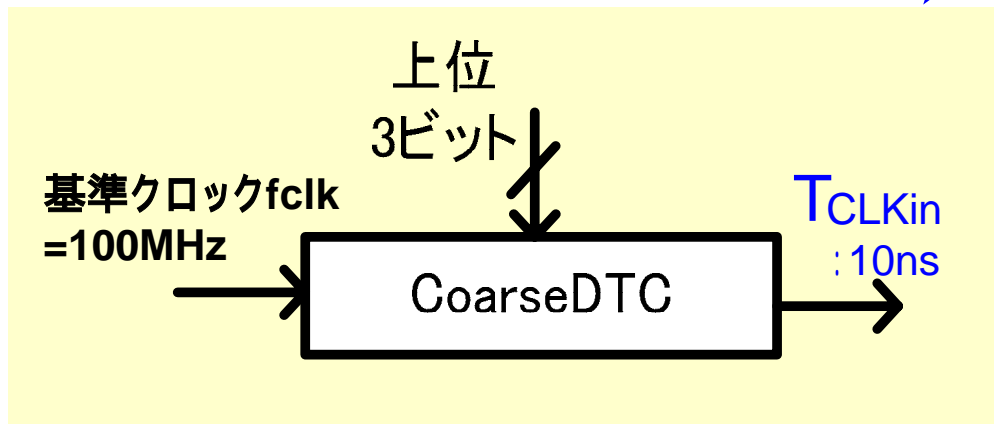
CoarseDTC

■分解能: 3bit

$$\frac{1/12.5[\text{MHz}]}{1/100[\text{MHz}]} = \frac{80\text{ns}}{10\text{ns}} = 2^3 : 3[\text{bit}]$$

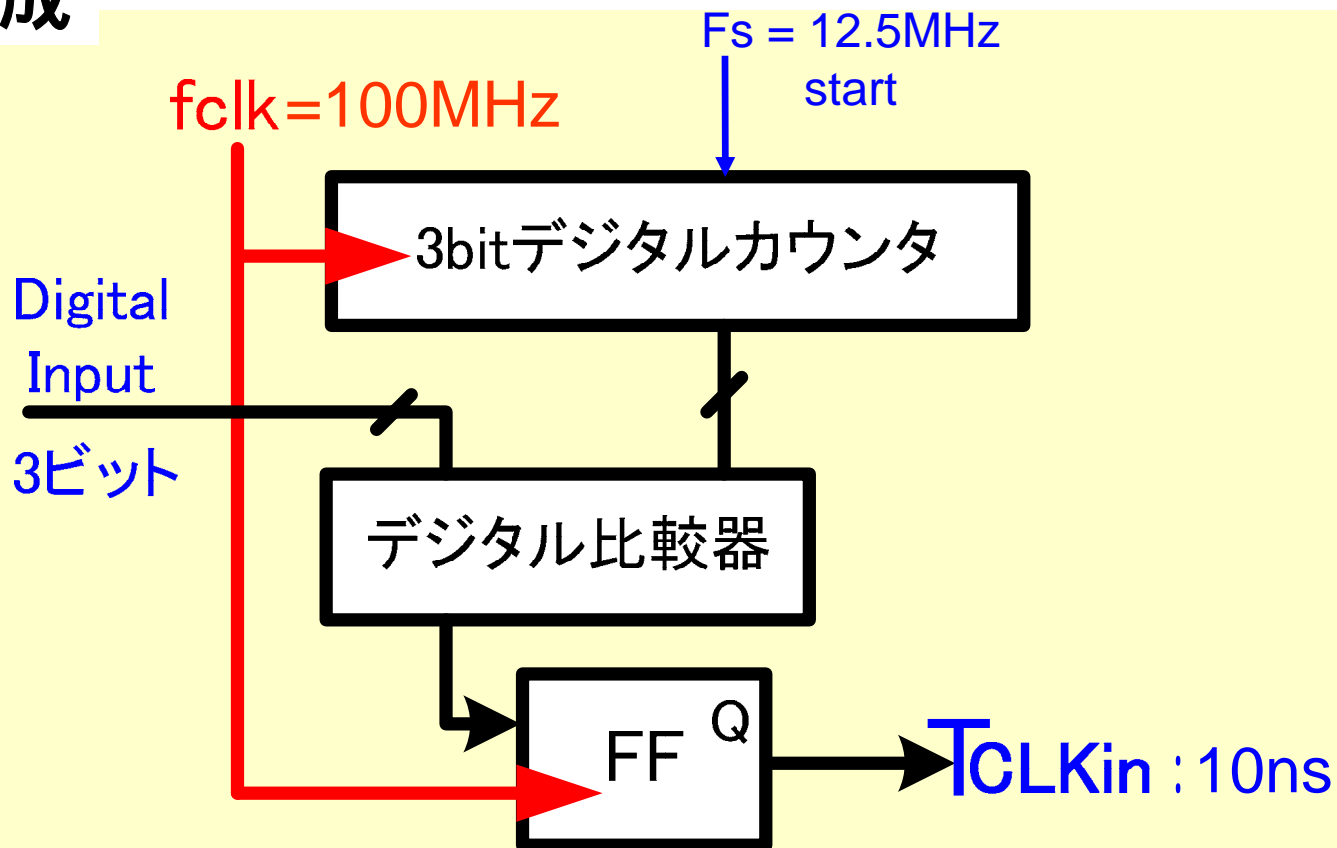
■時間分解能:

$$80\text{ns} / 2^3 = 10\text{ns} (\text{上位LSB})$$



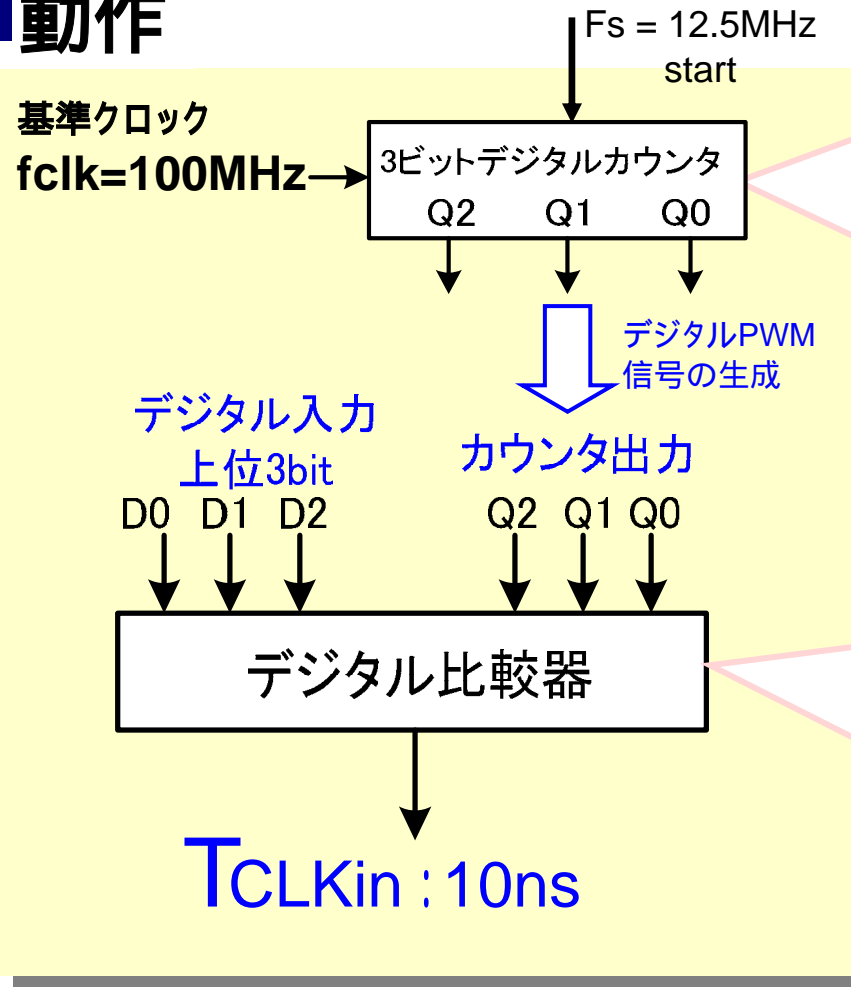
CoarseDTCの構成

■構成

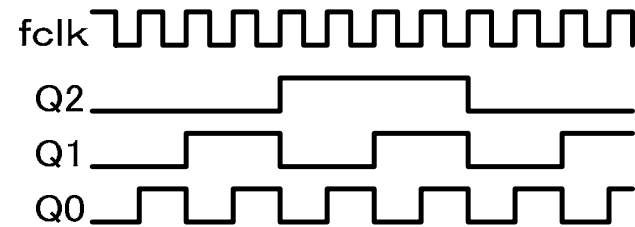


CoarseDTCの動作

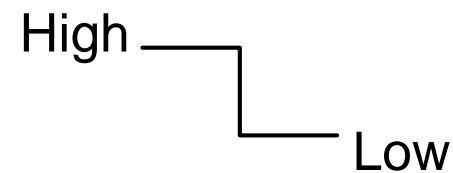
■動作



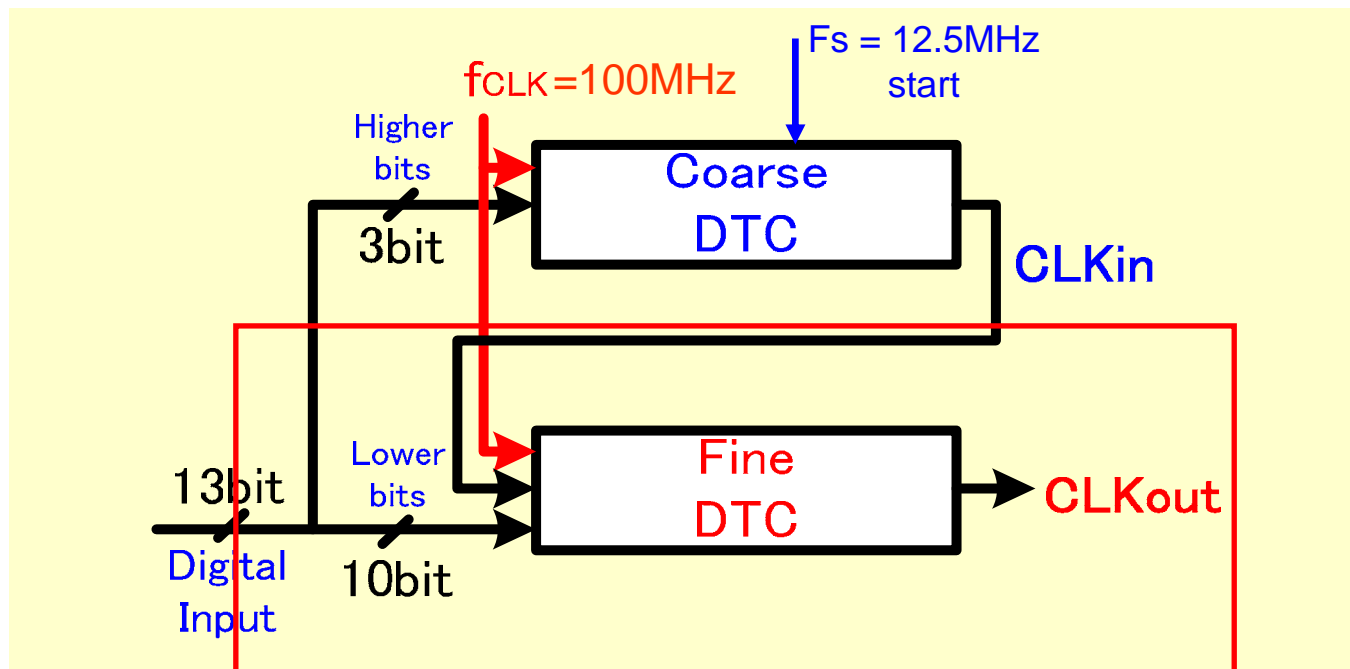
タイミングチャート



比較一致: CLKin生成
FineDTCに入力



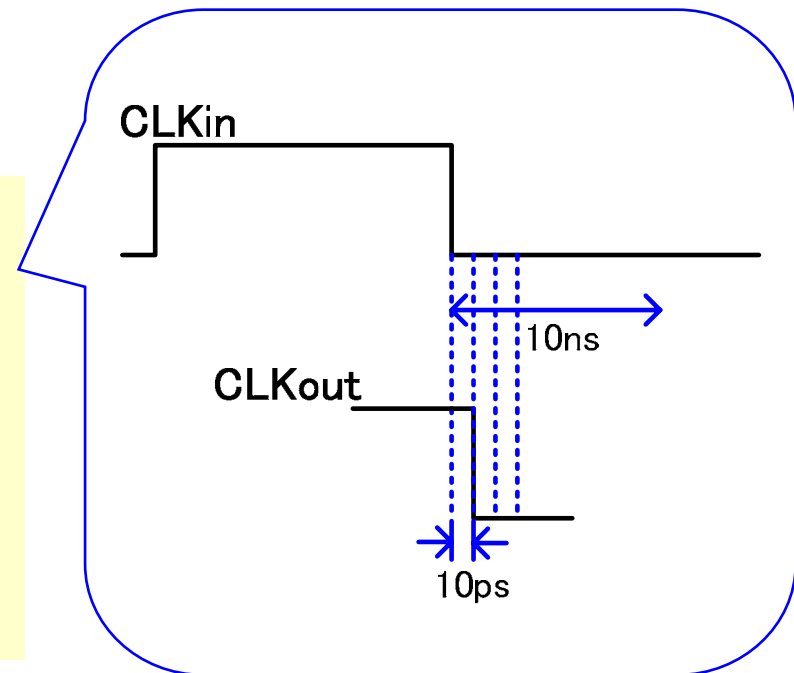
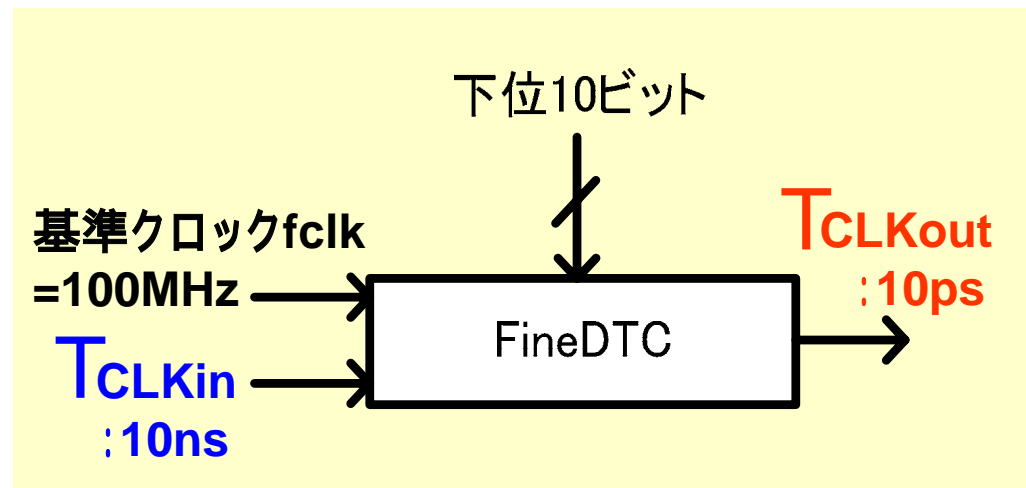
■ FineDTCについて



FineDTCの仕様

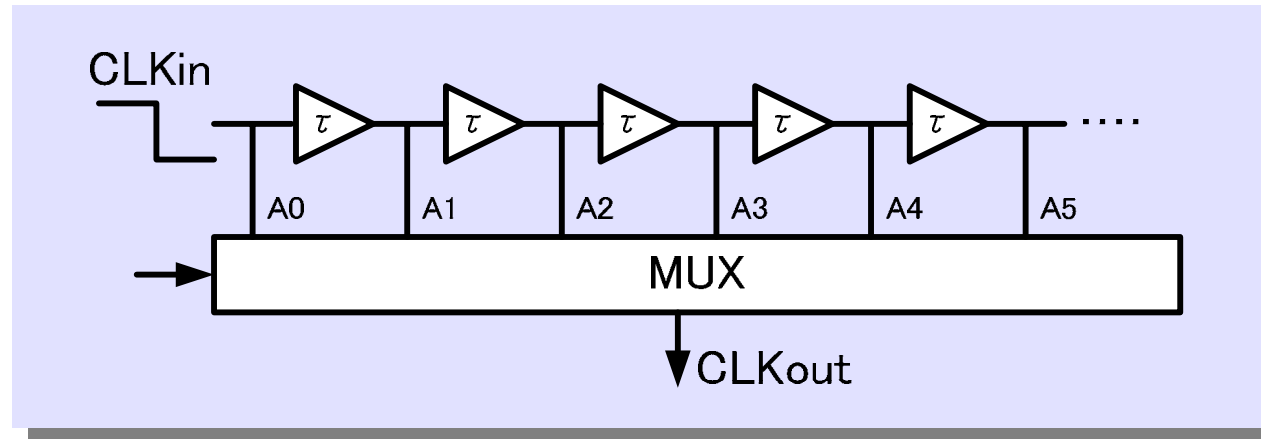
■FineDTC

- 分解能: 10bit
- 時間分解能: 10ps



FineDTC (従来の構成と問題点)

■構成



■問題点

■バッファ数: 大 (10bit設計 1023個)

■最小時間分解能


・バッファのゲート遅延: $\approx 10\text{ps}$

・半導体のプロセス性能に依存 (ゲート遅延によって高時間分解能を得る)

消費電力 \times ゲート遅延 = 一定

回路規模: 大、一つあたりの遅延量: 小

↳ 消費電力: とても大きい



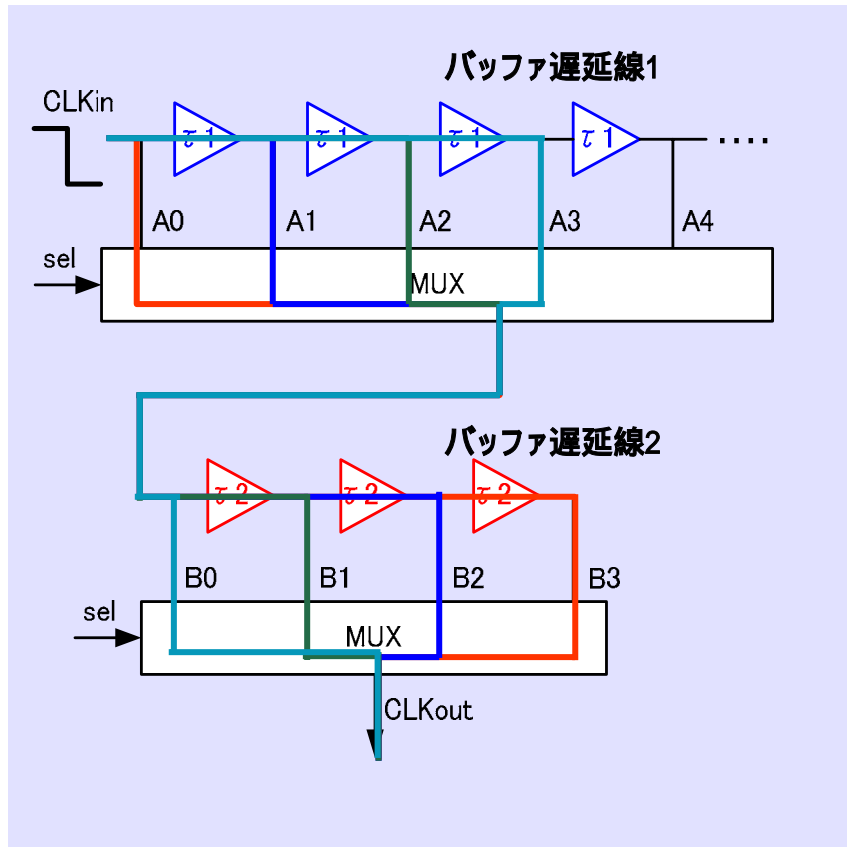
FineDTC (提案構成)

- 試行錯誤的回路
- システムチェック回路

2段バッファ構成FineDTCの動作

提案1) 試行錯誤的回路

■動作



(A0, B3) ... 基準

(A1, B2) ... 1- 2 =

(A2, B1) ... 2 1-2 2 = 2

(A3, B0) 3 1-3 2 = 3

(A1, B3) 1 = 4

(A2, B2) 2 1- 2 = 1+

(A3, B1) 3 1-2 2 = 1+2

(A4, B0) 4 1-3 2 = 1+3

(A2, B3) 2 1 (1 = 4)

(A3, B2) 3 1- 2 = 2 1+

(A4, B1) 4 1-2 2 = 2 1+2

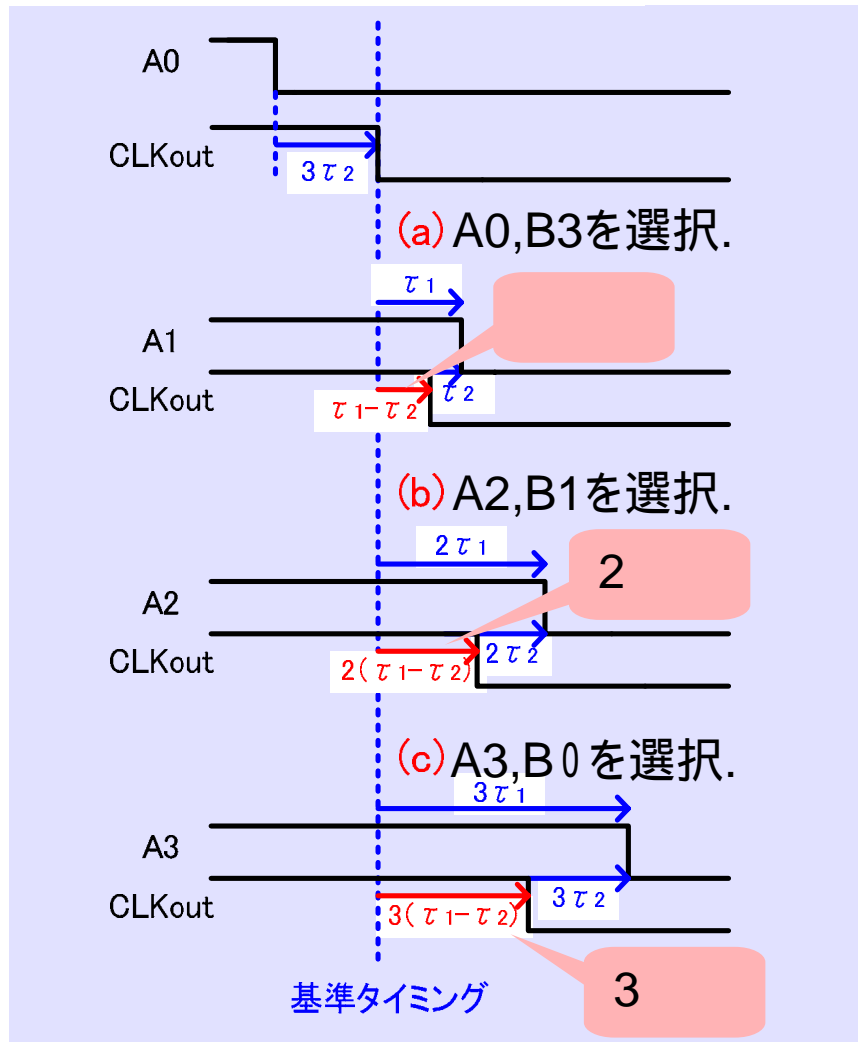
(A5, B0) 5 1-3 2 = 2 1+3

⋮

2段バッファ構成FineDTCの動作

提案1) 試行錯誤的回路

■ タイミングチャート



$$= 1 - 2$$

■ 時間分解能:

一つのバッファの
ゲート遅延量より小さくなる

バッファ単体の遅延量: 大
全体の時間分解能: 高精度

MUXの設計

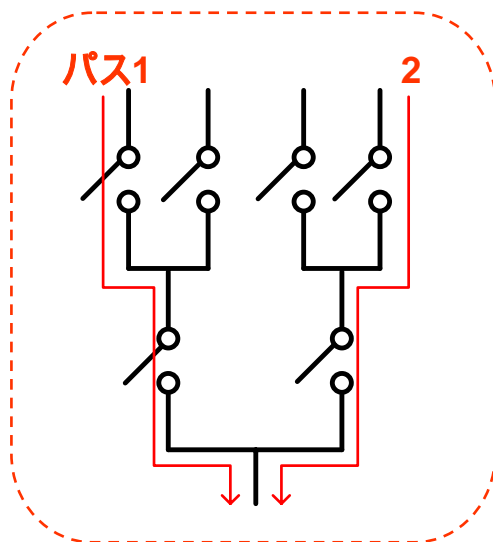
提案1) 試行錯誤的回路

■動作

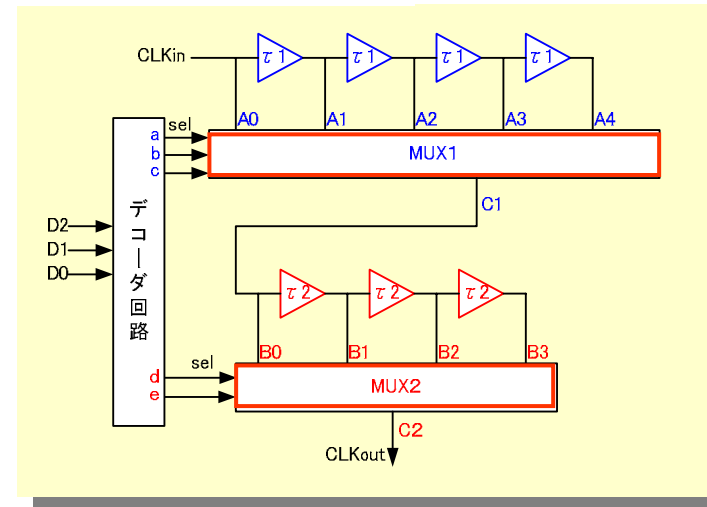
バッファの選択経路順を与える。

■設計

- ・レイアウト: 等長配線
- ・遅延差ゼロ



■3bitDTC回路図



■真理値表

■MUX 1

a	b	c	C1
0	0	0	A0
0	0	1	A1
0	1	0	A2
0	1	1	A3
1	0	0	A4

■MUX 2

d	e	C2
0	0	B3
0	1	B2
1	0	B1
1	1	B0

デコーダ回路の設計

提案1) 試行錯誤的回路

■動作

MUXへバッファの
選択アルゴリズムを与える。

■真理値表

	D2	D1	D0	a	b	c	d	e
A0 B3	0	0	0	0	0	0	0	0
A1 B2	0	0	1	0	0	1	0	1
A2 B1	0	1	0	0	1	0	1	0
A3 B0	0	1	1	0	1	1	1	1
A1 B3	1	0	0	0	0	1	0	0
A2 B2	1	0	1	0	1	0	0	1
A3 B1	1	1	0	0	1	1	1	0
A4 B0	1	1	1	1	0	0	1	1

■設計(論理式)

$$a = D2 \cdot D1 \cdot D0$$

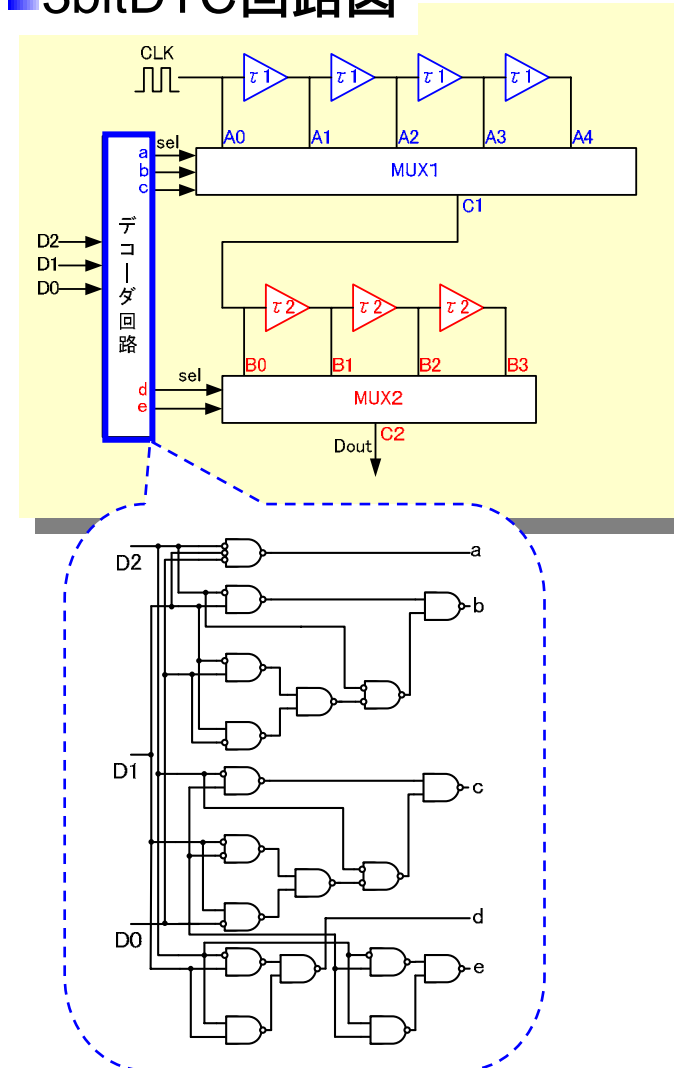
$$b = \overline{D2} \cdot D1 + D2(\overline{D1} \cdot D0 + D1 \cdot \overline{D0})$$

$$c = \overline{D2} \cdot D1 + D2(\overline{D1} \cdot \overline{D0} + D1 \cdot \overline{D0})$$

$$d = \overline{D2} \cdot D1 + D2 \cdot D1$$

$$e = \overline{D2} \cdot D0 + D2 \cdot D0$$

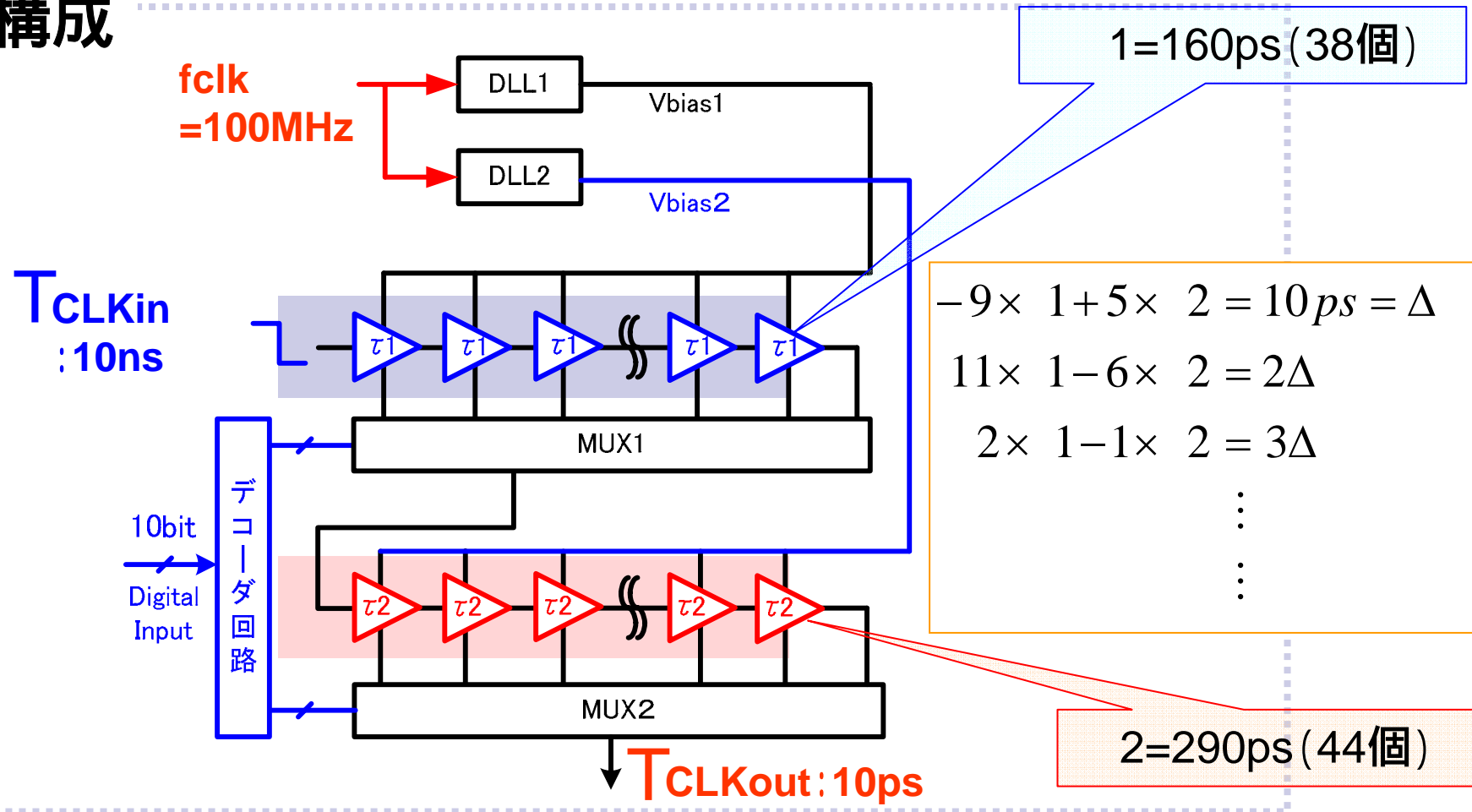
■3bitDTC回路図



2段バッファ構成FineDTCの新規提案

提案2) システムチック回路

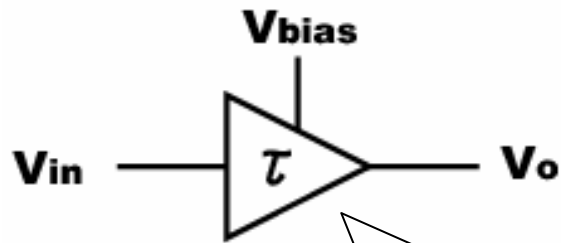
構成



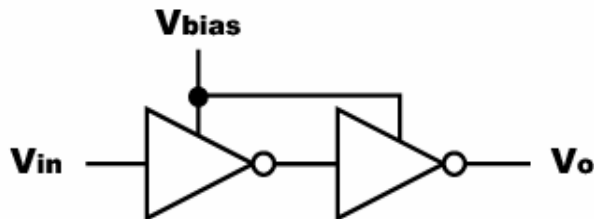
バッファの構成

提案2) システムチック回路

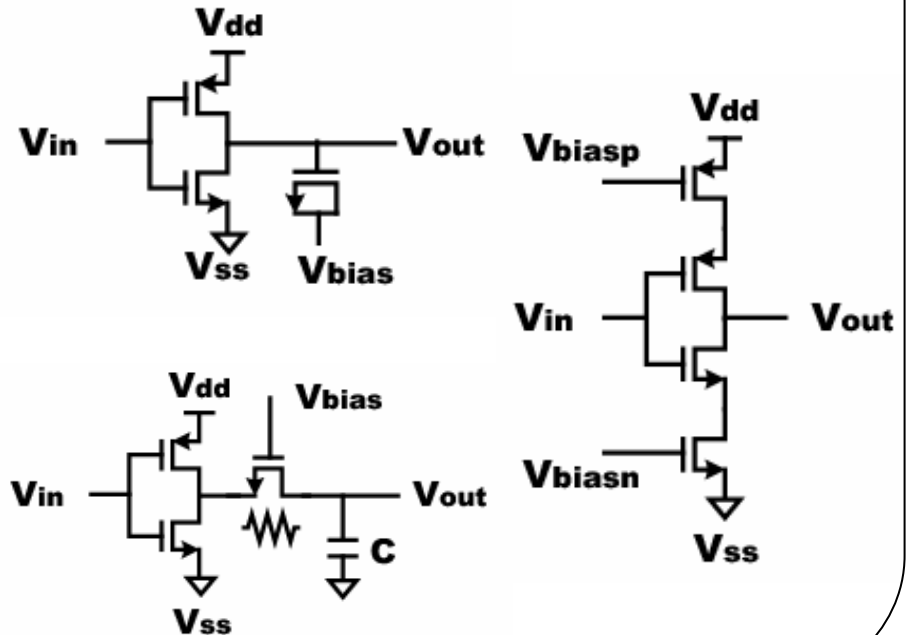
■ FineDTCに使うバッファ



■ バッファ遅延線バッファ回路 V_{bias} でインバータ遅延量を調整



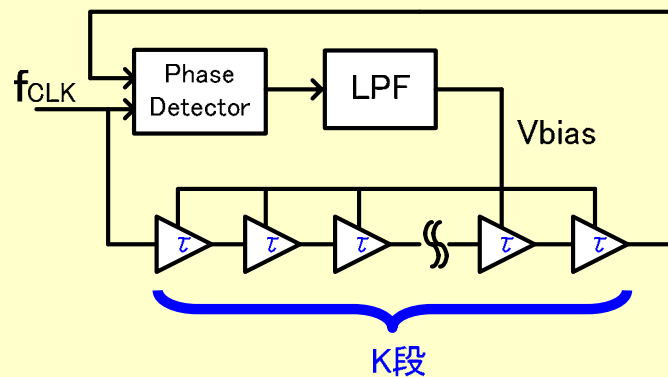
■ バイアス制御インバータ回路



FineDTCのDLL設計

提案2) システムチック回路

■構成



K段のバッファでfclkが1周期遅れるようにKを決める

$$K = \frac{1}{f_{clk}}$$

■動作

2つのゲート遅延の自動制御・調整

■設計

■DLL1 ($t_1=160\text{ps}$)

$$K1 = \frac{1}{1 \cdot f_{clk}} = \frac{1}{160 \times 10^{-12} \times 10^8} = 63\text{段}$$

■DLL2 ($t_2=290\text{ps}$)

$$K2 = \frac{1}{2 \cdot f_{clk}} = \frac{1}{290 \times 10^{-12} \times 10^8} = 34\text{段}$$

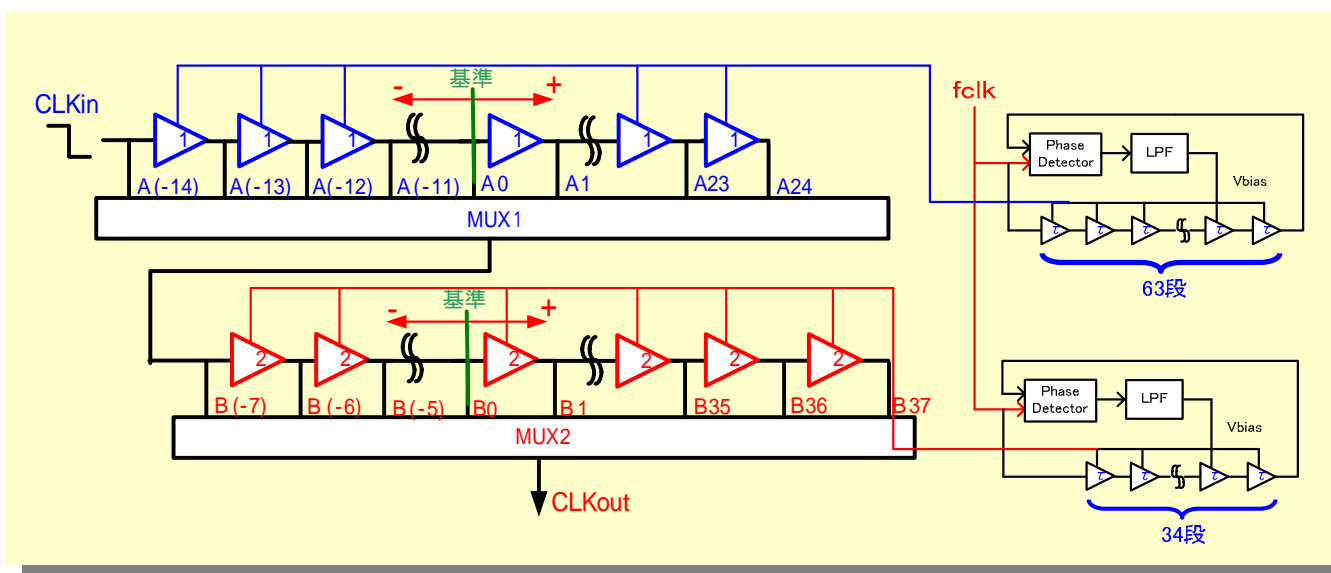
■DLLバッファ

DTCのバッファからなる回路を使用

2段バッファ構成FineDTC

提案2) システムチック回路

■ 提案回路構成図



■ 比較

従来: 1023個 (=10ps)

提案: 82個 (1: 160ps, 2: 290ps)

$$-9 \times 1 + 5 \times 2 = 10ps = \Delta$$

$$11 \times 1 - 6 \times 2 = 2\Delta$$

$$2 \times 1 - 1 \times 2 = 3\Delta$$

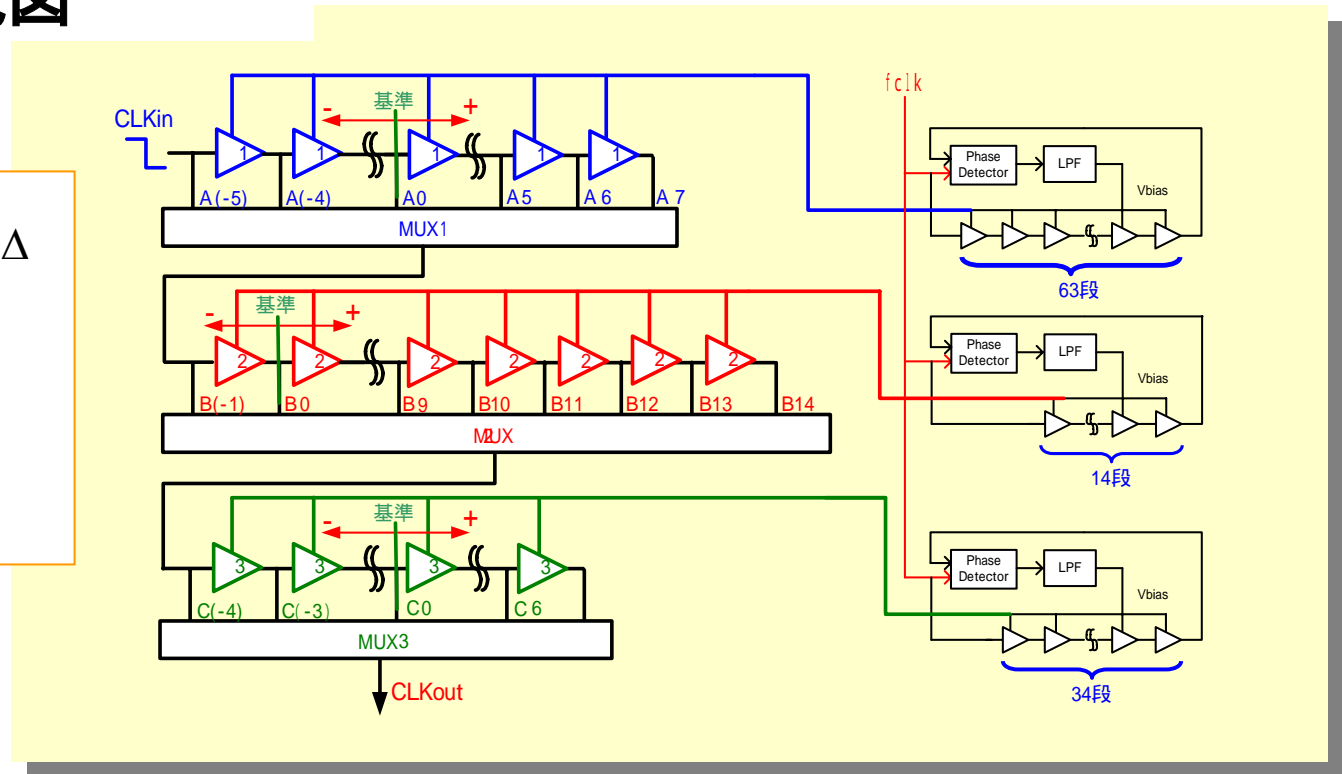
⋮
⋮

3段バッファ構成10bitFineDTCの新規提案

提案2) システムチック回路

■ 提案回路構成図

$$\begin{aligned}
 1 \times 1 - 1 \times 2 + 2 \times 3 &= 10ps = \Delta \\
 1 \times 1 + 1 \times 2 - 3 \times 3 &= 2\Delta \\
 2 \times 1 + 0 \times 2 - 1 \times 3 &= 3\Delta \\
 &\vdots \\
 &\vdots
 \end{aligned}$$



■ 比較

2段バッファ構成 : 82個 (1:160ps, 2:290ps)

3段バッファ構成 : 37個 (1:160ps, 2:730ps, 3:290ps)

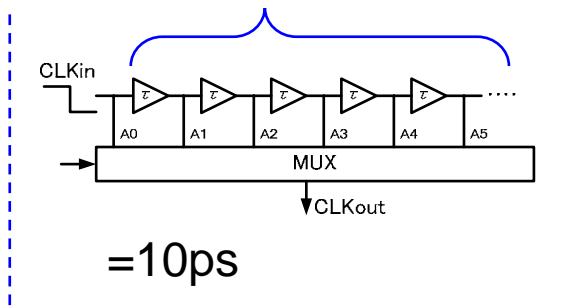
10bitFineDTCの設計結果

提案2) システムチック回路

■設計結果、比較

時間分解能: 10ps

□従来: 1023個

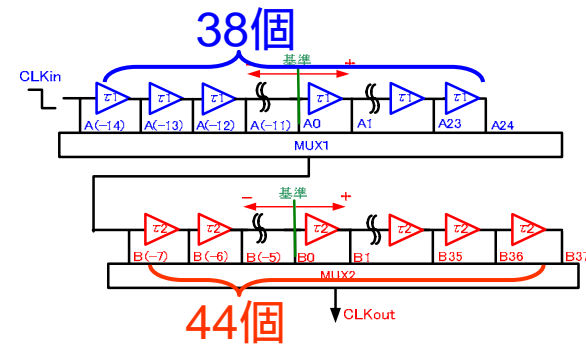


□2段

1=160ps

2=290ps

計: 82個



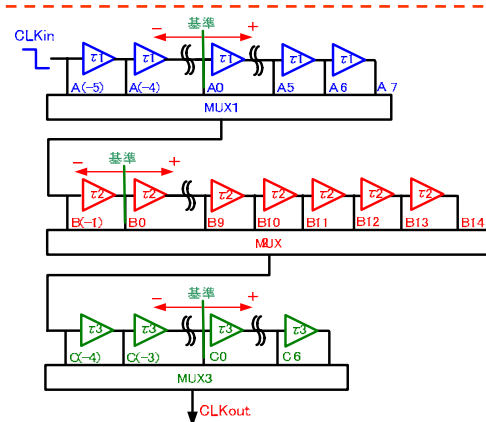
□3段

上段12個 (1=160ps)

中段15個 (2=730ps)

下段10個 (3=290ps)

計: 37個



設計効果

■従来FineDTCとの比較

分解能: 13bit

■時間分解能

10ps (従来150ps)

■回路規模(バッファ総数)

37個 (従来1023個)

■消費電力

従来の1/25

2つの効果

・バッファ数: 減 回路規模: 小
・ゲート遅延量: 大 一つあたりの消費電力: 低

⇒ DPWMの消費電力
激減



OUTLINE

- 研究背景と目的
- A/D変換器の低レテンシー化
- デジタルPWM (DPWM) 発生回路
- **DPWM設計アルゴリズム**
システムチック回路設計
- まとめと今後の課題

提案FineDTCの設計

～システムチック回路設計～

■ 試行錯誤的回路の改善点

■ バッファの選択が手計算のため単調で非効率

ビット数: 増 バッファ数: 大

■ 多段化への拡張: 難

3段、4段バッファ構成の設計への応用不可

■ 改良手法

■ 設計アルゴリズムの開発

・ 提案FineDTCの最適設計を理論化

・ 3,4段バッファ構成のシステムチック設計法を確立

・ バッファ数の削減と低消費電力化



- システムチック回路設計
 - 拡張ユークリッド互除法 -

拡張ユークリッド互除法とは？

■ 拡張ユークリッド互除法

x, y : 自然数、 l : x, y の最大公約数

$ax + by = l$ となる整数 a, b を求める手法

■ 例: $x=3, y=2, l=1$

→ $a \times 3 + b \times 2 = 1$

解

$(a, b) = (1, -1)$

$= (-1, 2)$

$= (3, -4)$

$= (-3, 5)$

⋮

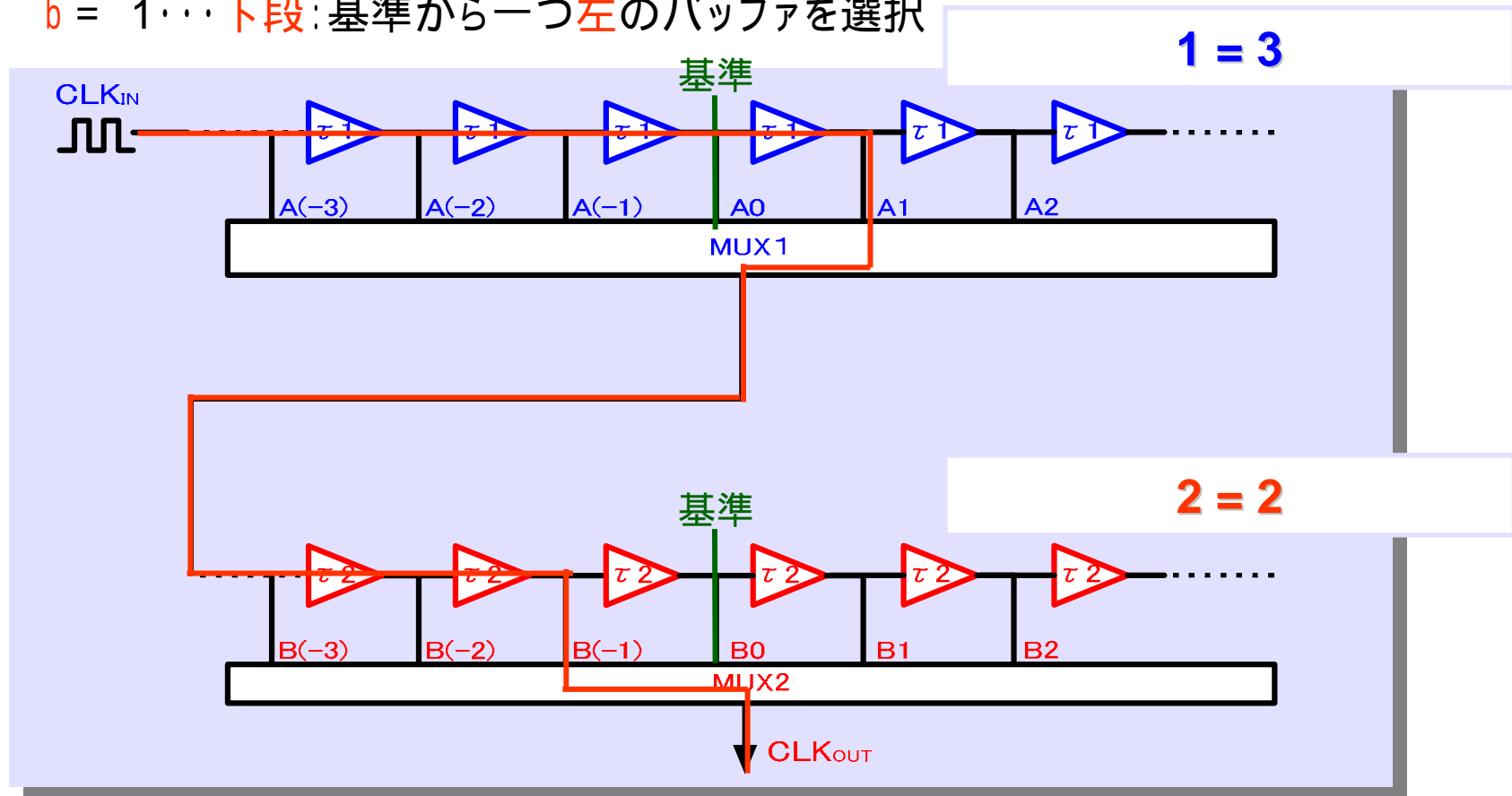
複数解: 存在

拡張ユークリッド互除法 -a,b-

■遅延 生成時のバッファ選択経路

$a = +1 \cdots$ 上段: 基準から一つ右のバッファを選択

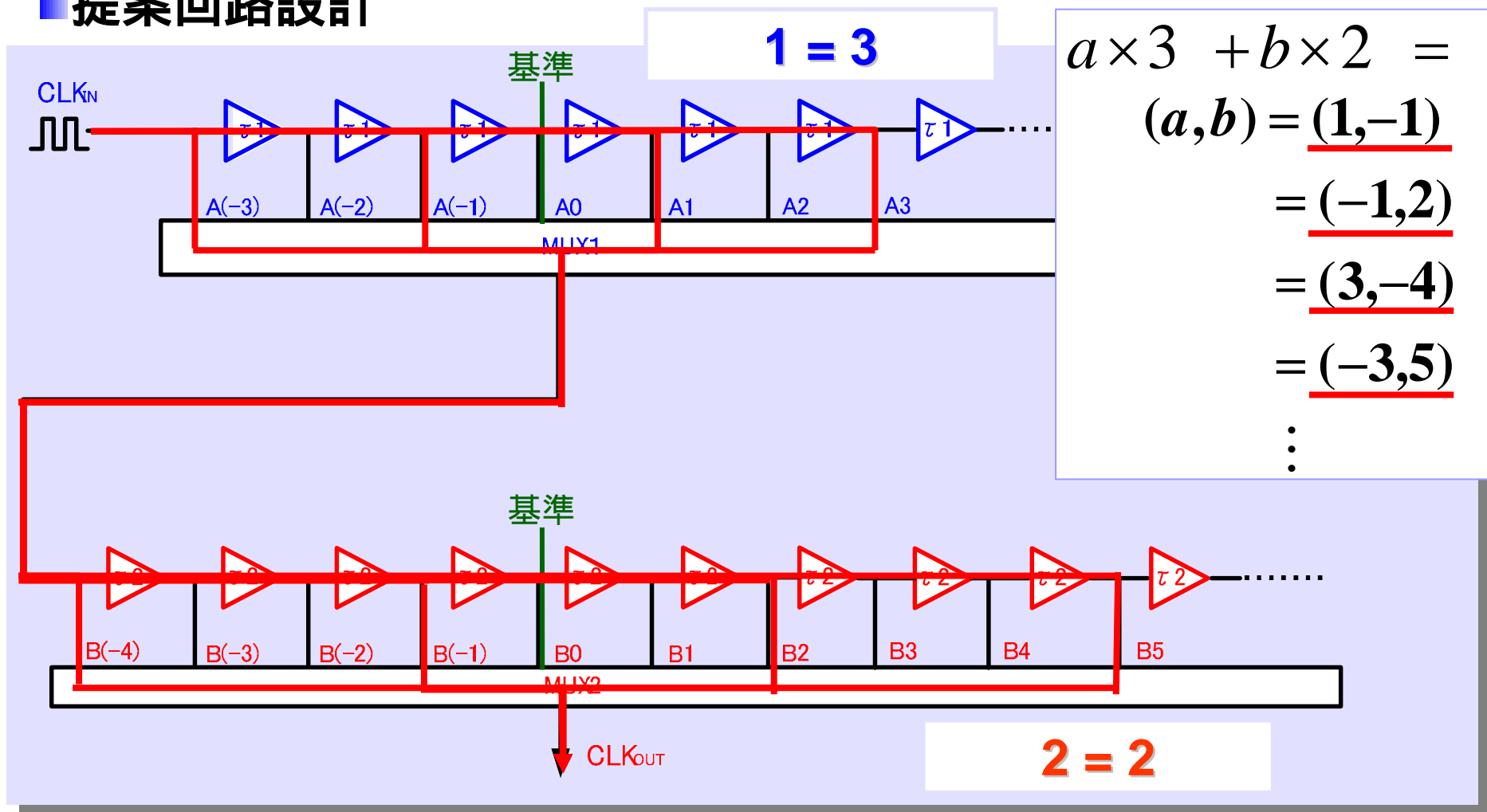
$b = 1 \cdots$ 下段: 基準から一つ左のバッファを選択



拡張ユークリッド互除法

-全ての多段バッファ構成回路に対応-

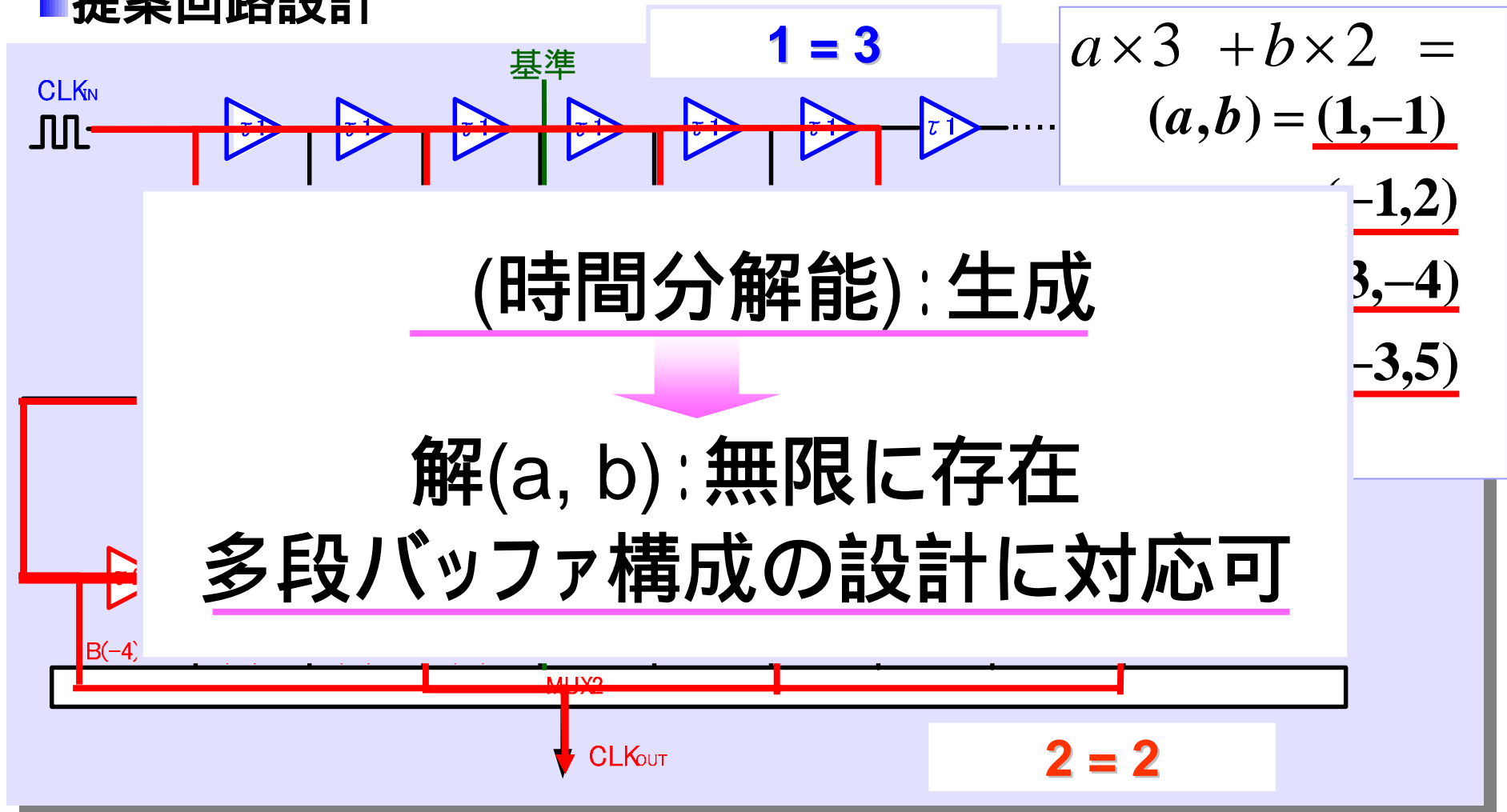
■提案回路設計



拡張ユークリッド互除法

-全ての多段バッファ構成回路に対応-

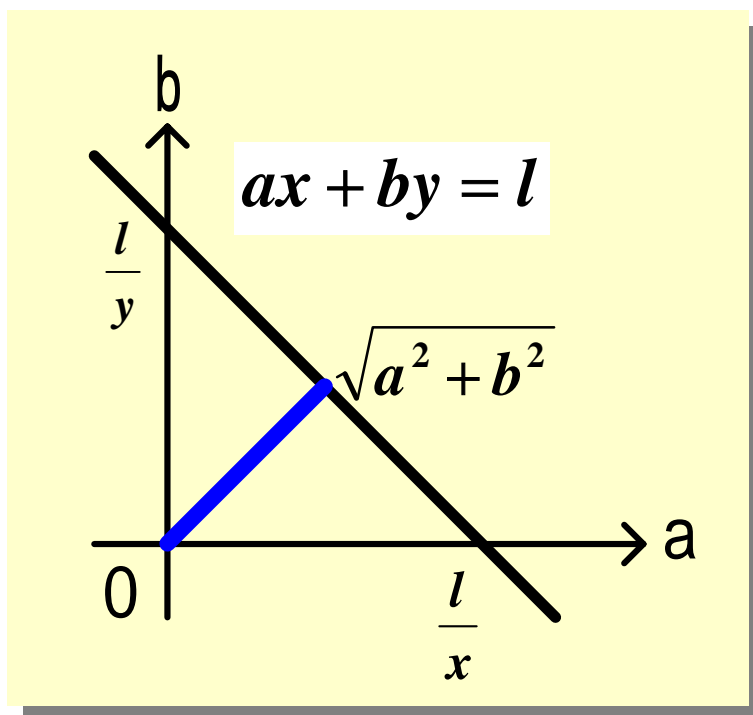
■提案回路設計



拡張ユークリッド互除法


■ a, b バッファ数

$|a|, |b|$: 最小の時 → バッファ数が最小となる回路構成



■ 原点からの距離最小

a, b をC言語プログラム計算

- 
- **2段バッファ構成FineDTC
(分解能10bit)への適用**

FineDTCの消費電力

■提案FineDTCの消費電力

CMOS回路の消費電力 $\propto \frac{1}{\text{処理時間}}$

$$\text{DTCの消費電力} = \text{ゲート数} + \left[\frac{\text{ゲート数}}{\text{ゲート遅延量}} \right]$$

MUX
DLL
(周辺回路)

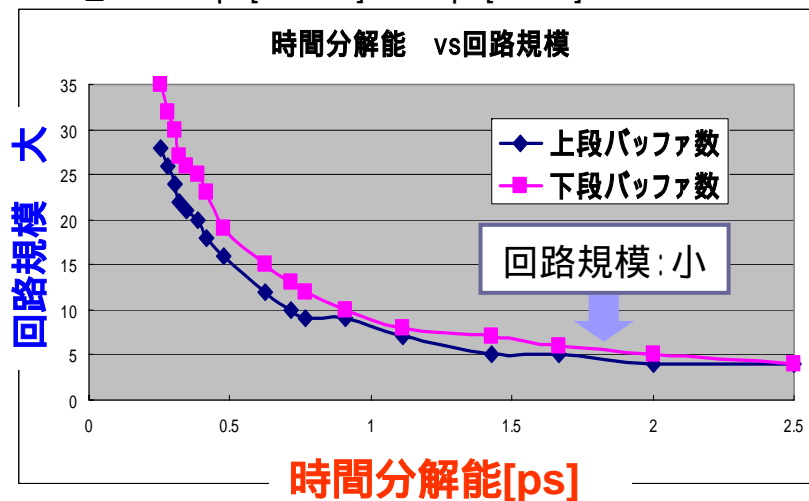
バッファ回路

一つのバッファ遅延量:大、バッファ数:減
相乗効果で消費電力:激減

時間分解能と回路規模

■時間分解能と回路規模の関係

4bit_ 1=10ps[130nm] 2=8ps[90nm]の時



■最適なFineDTCの設計法

■回路規模:小

ゲート遅延量 1, 2:大

1/ , 2/ :大

トレードオフの関係

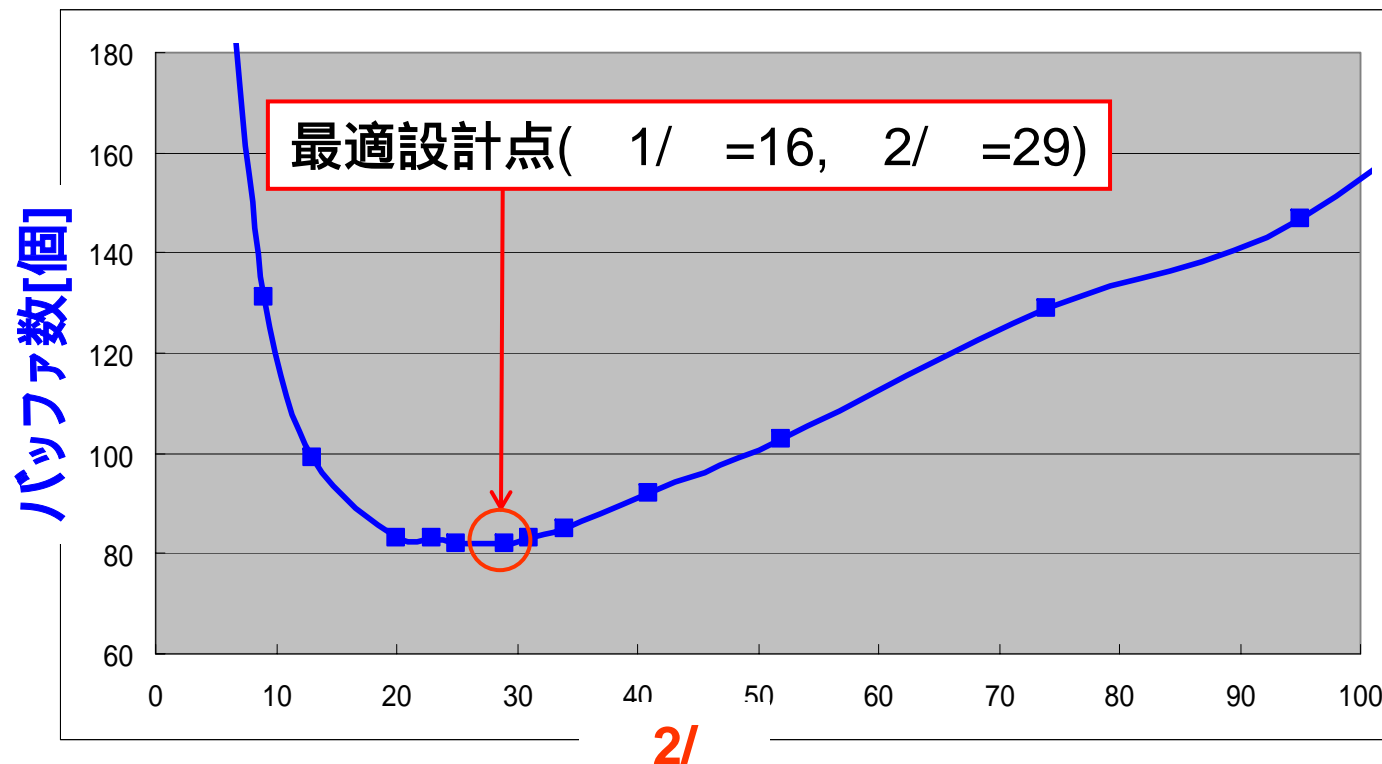
提案する設計アルゴリズムの適用

システムチック回路

2段バッファ構成(分解能10bit)

■ FineDTCの設計ポイントを探す

- 1/ (=3 ~ 67), 2/ (=5 ~ 121)時のバッファ数の変化
- 1: 2=1:1.8



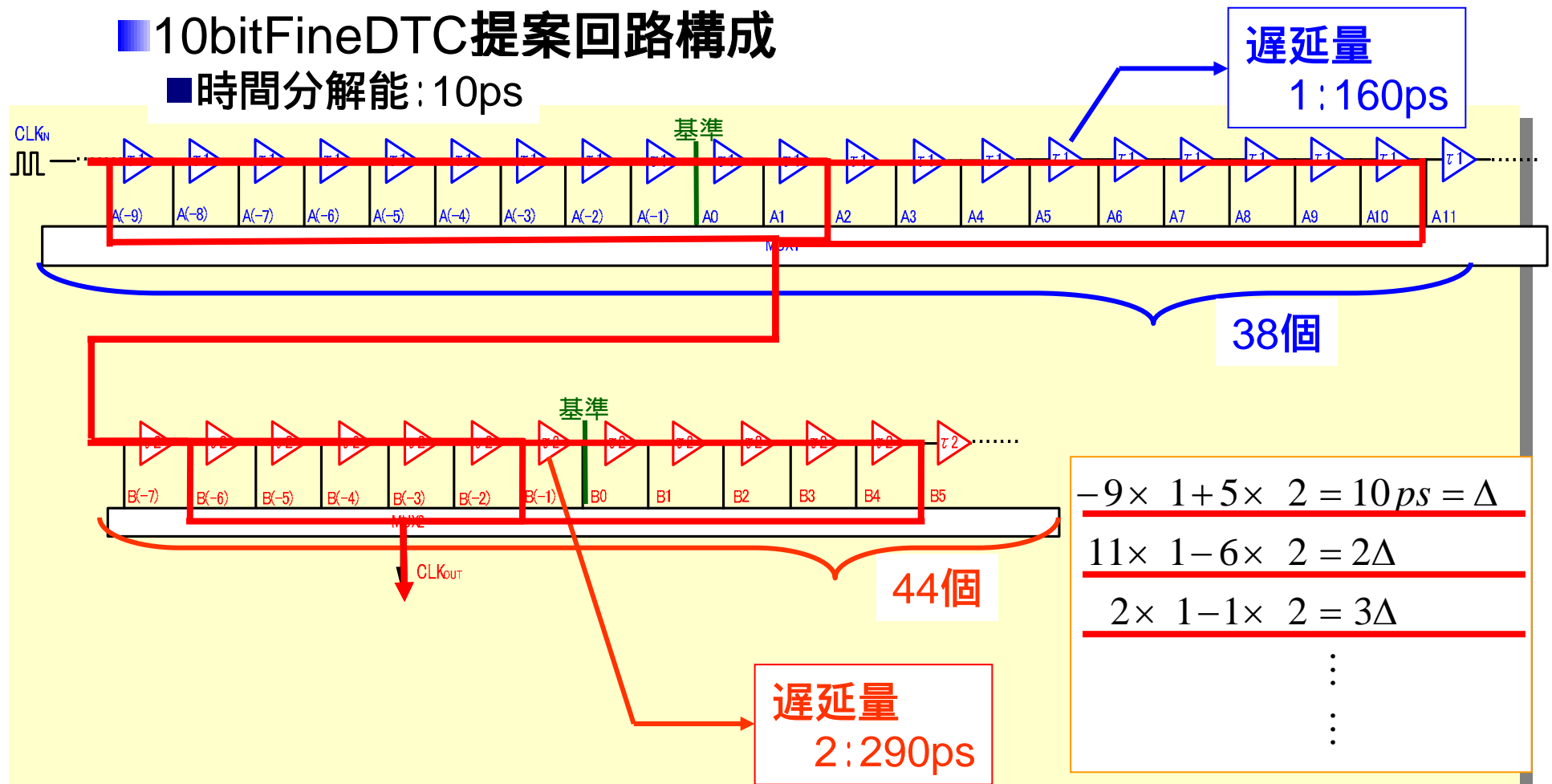
提案する設計アルゴリズムの適用

システムチック回路

2段バッファ構成(分解能10bit)

10bitFineDTC提案回路構成

時間分解能: 10ps

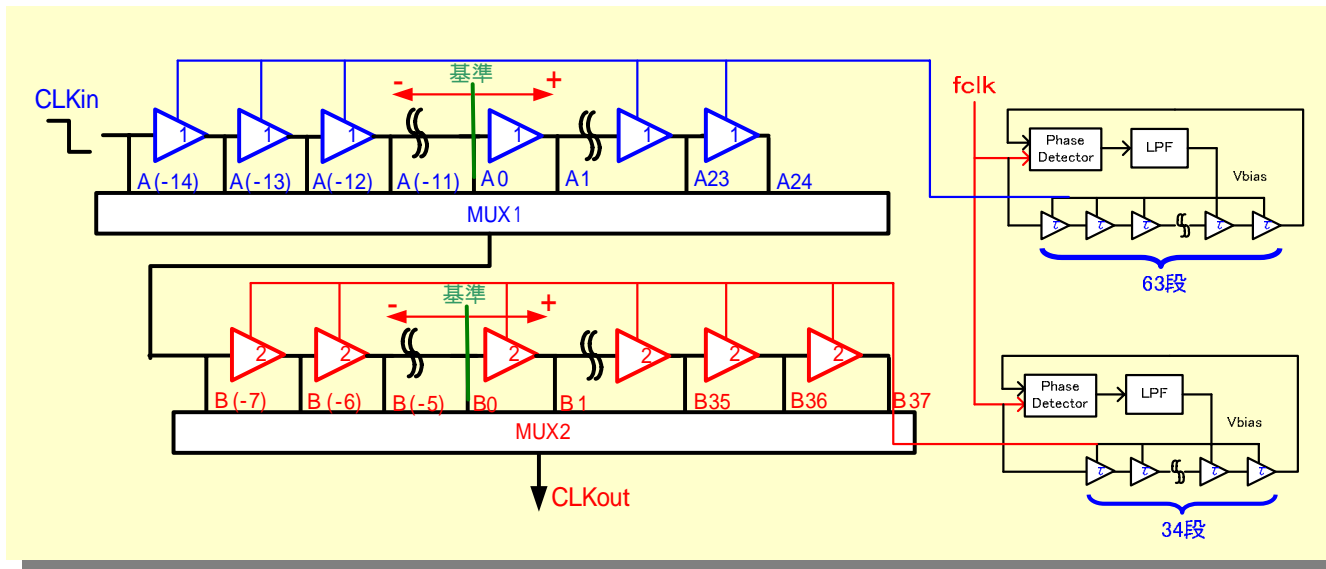


2段バッファ構成FineDTC

システムチック回路

2段バッファ構成(分解能10bit)

■提案回路構成図



■比較

従来: 1023個 (=10ps)


提案: 82個 (1: 160ps, 2: 290ps)

$$-9 \times 1 + 5 \times 2 = 10ps = \Delta$$

$$11 \times 1 - 6 \times 2 = 2\Delta$$

$$2 \times 1 - 1 \times 2 = 3\Delta$$

⋮
⋮

- 
- **3段バッファ構成FineDTC
(分解能10bit)への適用**

3段バッファ構成へ適用

- 3段バッファ構成の設計: 従来では困難
提案アルゴリズムにより適用可能

- 方法

$$ax + by + cz = l$$

- 3bitFineDTCの設計

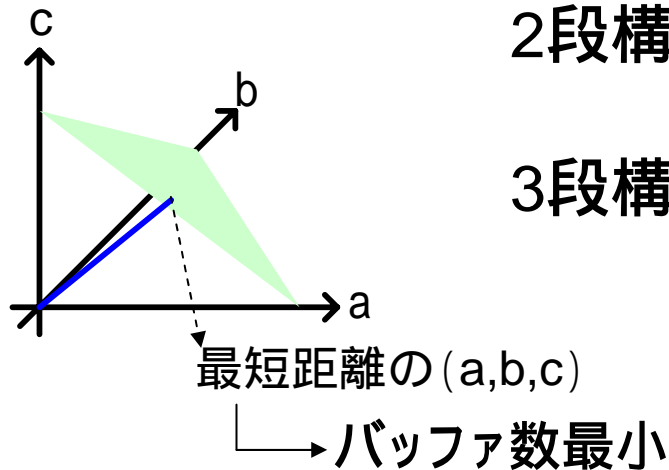
$$\cdot 1/ = 5, \quad 2/ = 4, \quad 3/ = 3$$

- 計算結果

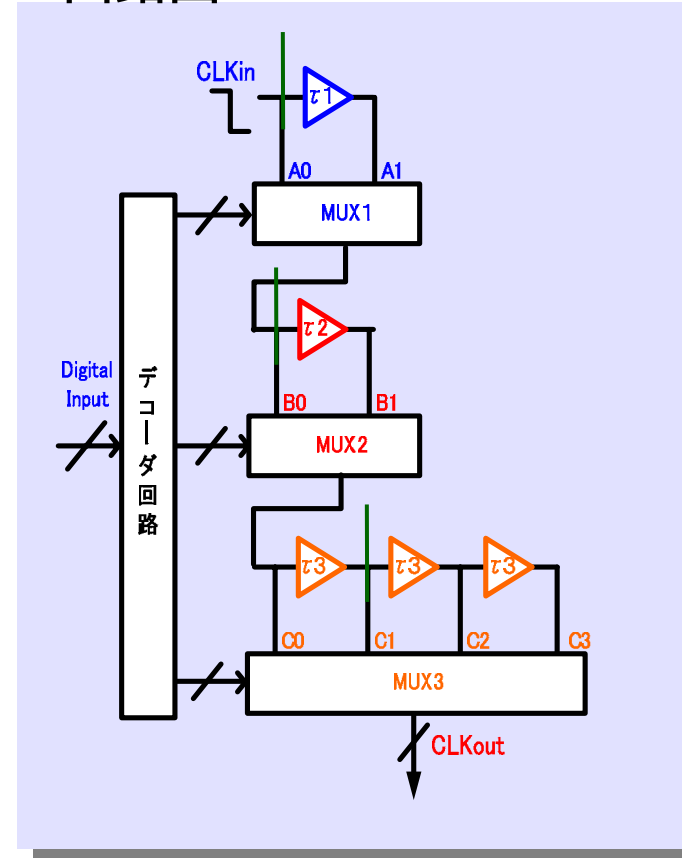
- バッファ数

2段構成: 6個

3段構成: 5個



- 回路図



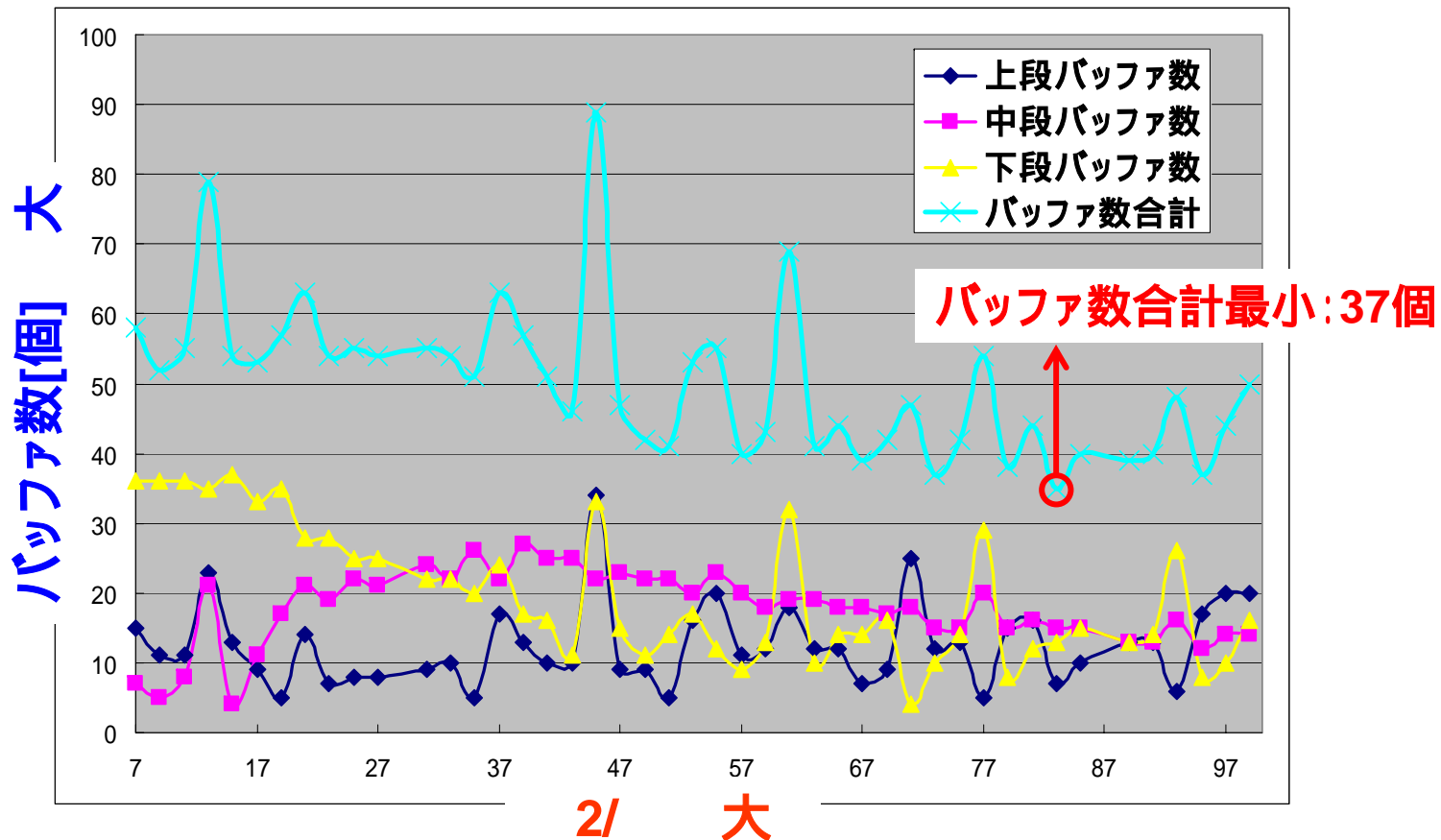
提案する設計アルゴリズムの適用

システムチック回路

3段バッファ構成(分解能10bit)

■ FineDTCの設計ポイントを探す

■ 1/ =16, 2/ =7 ~ 99 3/ =29のバッファ数の変化



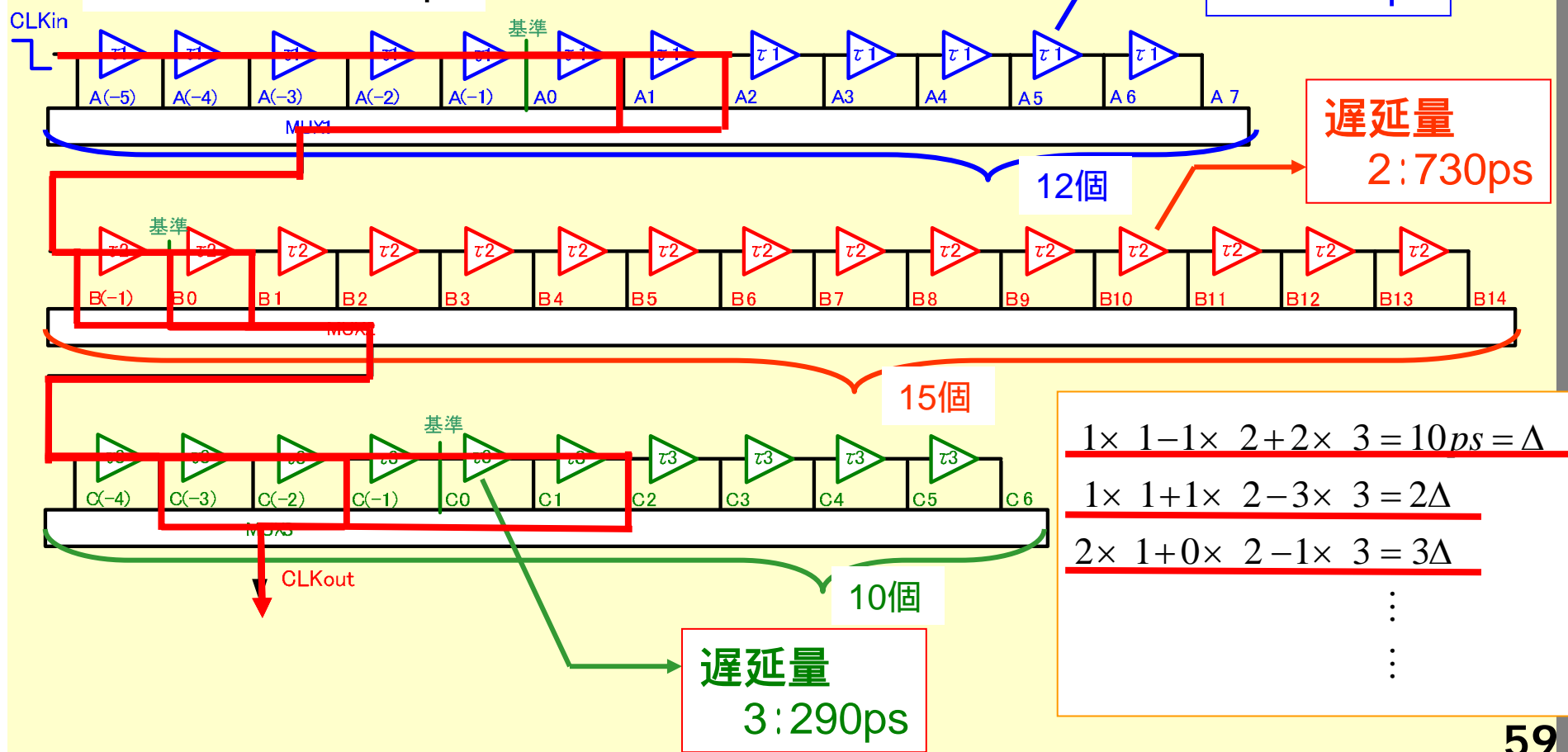
提案する設計アルゴリズムの適用

システムチック回路

3段バッファ構成(分解能10bit)

10bitFineDTC提案回路構成

■時間分解能: 10ps



多段バッファ構成の効果

■ 4段バッファ構成

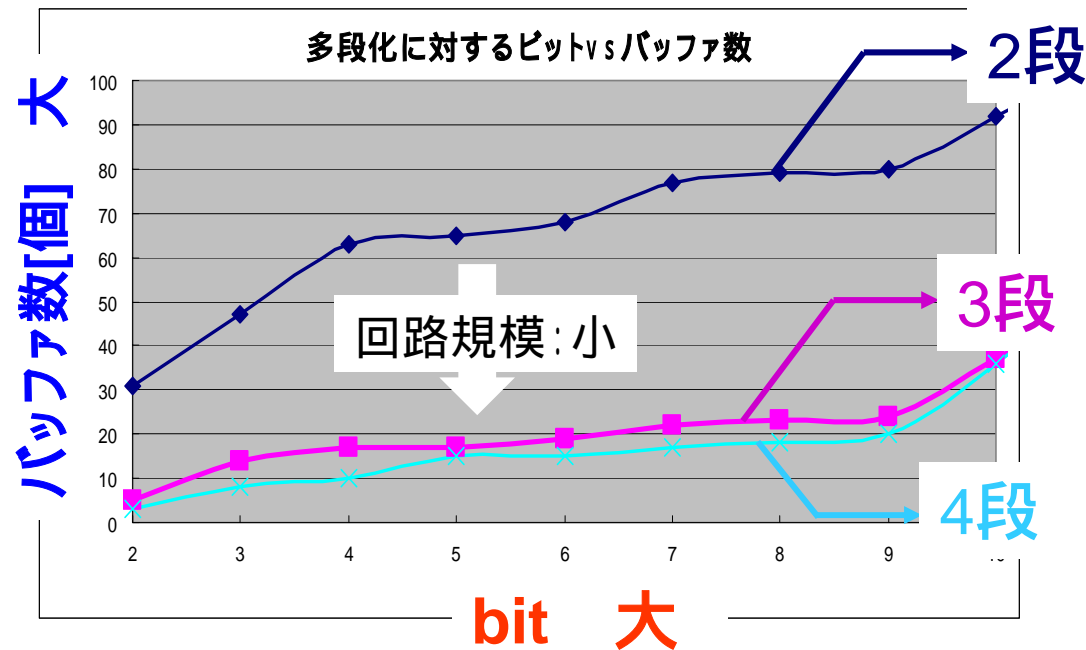
- 1 = 5
- 2 = 23
- 3 = 47
- 4 = 73

■ 3段バッファ構成

- 1 = 5
- 2 = 23
- 3 = 73

■ 2段バッファ構成

- 1 = 5
- 2 = 73



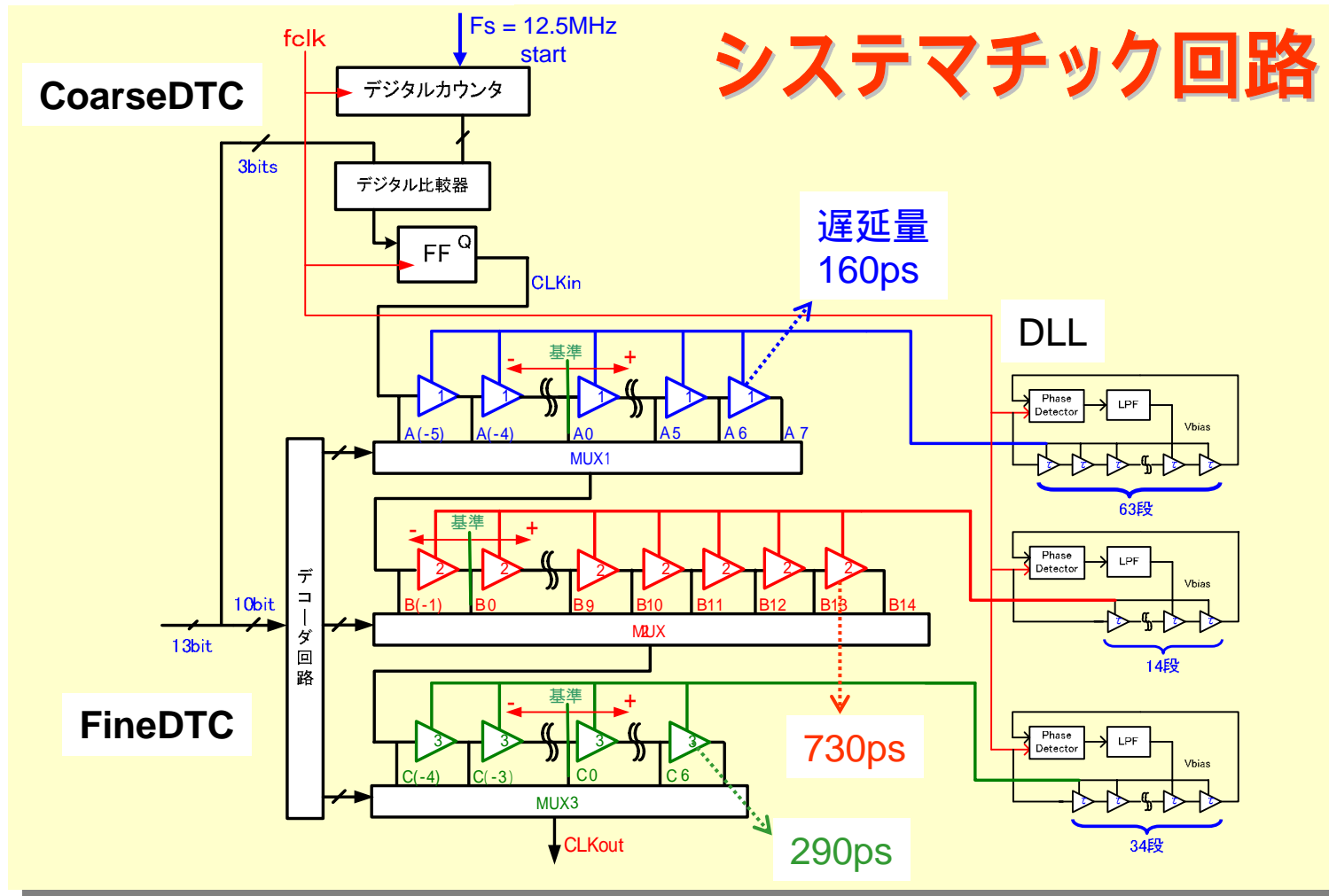
■ 結果

- ・回路規模: 2段 > 3段 > 4段
- ・∴ 段数増のデメリット MUXやDLL等周辺回路増

3段バッファ構成 **ベスト設計**

13bitDPWM発生回路

■提案全体回路構成





OUTLINE

- 研究背景と目的
- A/D変換器の低レテンシー化
- デジタルPWM(DPWM)発生回路
- DPWM設計アルゴリズム
- **まとめと今後の課題**



まとめ

- デジタル電源でのA/D変換器の低レーテンシー化
- D/PWM発生回路の新アーキテクチャ
 - 時間分解能: 10ps, 13bit
 - バッファ数: **37個** (従来 1023個)
 - ゲート遅延量:
160ps (上段), 730ps (中段), 290ps (下段) (従来10ps)
 - 消費電力: 従来**の1/25**
 - システムチック設計アルゴリズム

今後の課題

- FPGA等によるハードウェア実現
- 線形性の確保