

## Application Note

## マイコン + SDK 上の用の拡張 OSPI PHY チューニング アルゴリズム Sitara™ プロセッサ



Nikhil Jain, Aryamaan Chaurasia, Vaibhav Kumar, Soumya Tripathy, Brian Elies, Nuruddin Mahmood

## 概要

このアプリケーション ノートは、マイコン + SDK 11.02 の新機能である OSPI DQS PHY チューニング アルゴリズムの概要を示します。これは、AM243x、AM62x、AM62Ax、AM62Dx、AM62Px、AM64x、および AM275x デバイスなど、テキサス インストルメンツの Sitara™ プロセッサ ファミリの高速シリアル NOR および NAND フラッシュ アクセスを最適化するように設計されています。PHY チューニング プロセスは、Tx DLL、Rx DLL、および読み取り遅延を含む三つのタイミングパラメータを自動的にキャリブレーションします。これらのパラメータは、基板レベルの伝搬遅延、プロセス、電圧、および温度の変動を補償します。対象読者には、Sitara プロセッサ上で OSPI フラッシュ メモリを扱う組み込みソフトウェア エンジニア、システム アーキテクト、およびハードウェア設計者が含まれます。

## 目次

1 概要	2
2 用語	2
3 PHY のチューニング アルゴリズム	4
4 主要なチューニング パラメータ	5
4.1 非 DQS PHY チューニング アルゴリズムのパラメータ構成	5
4.2 DQS PHY チューニング アルゴリズムのパラメータ構成	5
5 PHY チューニング アルゴリズムの前提条件	6
5.1 ハードウェア要件	6
5.2 アタック ベクタ	6
5.3 合格領域と不合格領域の比較	6
5.4 マスタ モードとバイパス モードの比較	7
6 新しいチューニング アルゴリズムの必要性	8
6.1 温度変化	8
7 アルゴリズムの実装	9
7.1 DQS PHY のチューニング アルゴリズム	9
7.2 非 DQS PHY チューニング アルゴリズム	15
8 チューニングの拡張	19
8.1 調整時間最適化 – スキップ調整機能	19
8.2 ランタイム検証 – OTP の検証	19
9 まとめ	19
10 参考資料	19

## 商標

Sitara™ is a trademark of Texas Instruments.

すべての商標は、それぞれの所有者に帰属します。

## 1 概要

最新の組込みシステムでは、コード実行およびデータ保存のために外部フラッシュメモリへの高速かつ信頼性の高いアクセスが求められます。OSPI は、シリアル NOR および NAND フラッシュ デバイスとの高帯域通信を可能にし、シングルデータレート (SDR) およびデュアル データレート (DDR) モードを備え、8 レーン構成で高いデータレートをサポートしています。

このような高速動作では、信号タイミングが非常に重要です。基板配線長のわずかなばらつき、温度変化、電圧変動、および部品のばらつきにより、OSPI コントローラが誤ったタイミングでデータをサンプリングすると、データが破損する可能性があります。OSPI PHY (物理層) モジュールには、プログラマブルな遅延ラインが含まれており、信号タイミングを微調整できますが、最適な遅延設定を手動で見つけることは実用的ではありません。

遅延設定を決定するために、PHY チューニング アルゴリズムはパラメータ空間を検索して安定した動作点を特定します。このアプリケーション ノートでは、アルゴリズムの動作原理、チューニング成功に影響する要因、および DQS チューニング アルゴリズムで実行される手順について説明します。

## 2 用語

PHY:

OSPI ドライバの PHY (物理層) モードを指します。PHY モードでは、メモリ データ転送を管理するために専用のタイミング回路を使用します。このモードでは、各リファレンス クロック サイクルは標準転送用に完全な 1 メモリ クロック サイクル、ダブルスピード転送用に半サイクルを生成します。このシステムは、内部信号またはメモリ チップからの外部フィードバックを使用して、4 種類のタイミング設定を提供します。

PHY が有効化されている場合、入力クロック分周器はバイパスされます。その結果、実効周波数は入力クロック周波数となります。PHY チューニング アルゴリズムは、rxDLL、txDLL、Read Delay を変化させることによって最適なチューニングポイントを計算します。詳細については、[こちら](#)をご覧ください。

QSPI:

クワッド シリアル ペリフェラル インターフェイス - シリアル データ転送用に 4 つのデータライン (DQ0 ~ DQ3) を使用する拡張 SPI バリエーションです。さまざまな転送フェーズのためにシングル / デュアル / クワッドのモードをサポートしており、標準の SPI に対して最大 4 倍の帯域幅向上を実現すると同時に、下位互換性を維持します。

OSPI:

オクタール シリアル ペリフェラル インターフェイス - シリアル データ転送用に 8 つのデータライン (DQ0 ~ DQ7) を使用する高度な SPI バリエーション。すべての QSPI モードとオクタール モードをサポートしており、より広い帯域幅に対応しています。ソース同期データ キャプチャ用の DQS (データ ストローブ) 信号の有無にかかわらず動作可能です。

SDR:

シングル データレート モードは、クロック信号のシングル エッジでデータを転送し、データラインごとのクロックサイクルごとに 1 ビットを送信します。これは、中程度の速度で優れた信頼性を実現する、よりシンプルでより実績のあるクロック供給方式です。8 本のデータラインを持つオクタール SDR モードでは、理論上の最大データレートはクロック サイクルあたり 8 ビットです。

DDR:

ダブル データレート モードは、クロック信号の立ち上がりエッジと立ち下がりエッジの両方でデータを転送し、SDR モードと比較して、データ スループットが実質的に 2 倍になります。データラインが 8 つのオクタール DDR モードでは、データはクロック サイクルごとに 16 ビット (エッジごとに 8 ビット × 2 つのエッジ) 転送されます。

プロトコル (コマンド - アドレス - データ) :

プロトコル モードのフォーマットは WR-WR-WR であり、最初の WR はコマンドビットの幅とレートを、2 番目の WR はコマンド修飾ビットの幅とレートを、3 番目の WR はデータビットの幅とレートを表します。ビット幅 (W) は 1 ビットまたは 8 ビットが可能です。レート (R) は、SDR の場合は S、DDR の場合は D です。SDR は立ち上がりクロック エッジと立ち下がりクロック エッジの両方で同じ値を転送しますが、DDR は各エッジで異なる値を転送できます。

たとえば、1S-1S-1S は、すべての位相が 1 ビット幅 SDR を使用することを意味します。8D-8D-8D という表記は、すべての位相が 8 ビット幅 DDR を使用することを意味します。

#### DQS:

データ ストローブは、DDR モードにおけるフラッシュ読み出し動作中にフラッシュ デバイスから提供されるソース同期信号です。DQS のエッジは、有効なデータ ウィンドウの中心に整合されており、これによりコントローラは最適なタイミングでデータをサンプリングできます。DQS は DDR モードで使用され、SDR モードでは使用されません。

#### リファレンス クロック:

リファレンス クロックは、OSPI コントローラへの入力クロック信号です。これは通常、システム クロックによって提供されます。このクロックは分周され、フラッシュ デバイスに送信されるシリアル クロックが生成されます。

#### DLL:

遅延ロック ループは、は、信号タイミング制御のために高精度でプログラム可能な遅延を生成するデジタル回路です。DLL は、遅延素子のチェーンで構成されており、これらは通常インバータまたはバッファであり、それぞれの伝搬遅延は調整可能です。ループは、出力がフィードバック信号と一致するまで遅延チェーンを継続的に調整することで、リファレンスにロックします。OSPI PHY では、送信および受信のそれぞれに独立した DLL が用いられ、一つの遅延素子ステップの分解能で出力および入力信号のタイミングを制御し、セットアップ時間およびホールド時間をきめ細かく調整できます。

#### Tx DLL:

送信 DLL は PHY 内のプログラム可能な遅延ラインであり、コントローラがフラッシュに対して送信データおよびコマンドを出力するタイミングを調整します。また、セットアップ時間およびホールド時間の要件が満たされることを保証します。値の範囲は通常 0 ~ 127 です。

#### Rx DLL:

受信 DLL は、PHY 内のプログラム可能な遅延ラインであり、コントローラがフラッシュからの入力データをサンプリングするタイミングを調整します。DLL はサンプリング クロックまたは DQS 信号を遅延させ、有効データ ウィンドウの中心に整合させます。値の範囲は通常 0 ~ 127 です。

#### 読み取り遅延:

読み取り遅延は、は、データ キャプチャ タイミングに適用される追加のプログラム可能な遅延であり、リファレンス クロック サイクルで測定されます。これは、きめ細かな DLL 設定を補完する粗調整を提供します。値の範囲は通常 0 ~ 4 です。

#### OTP:

最適チューニングポイントは、チューニング アルゴリズムによって選択される Tx DLL、Rx DLL、および読み取り遅延の値の特定の組み合わせです。これは、すべての方向で最大のマージンを持つ安定した動作領域の中心を表します。

#### メタスタビリティ ギャップ:

メタスタビリティ ギャップは、均一に合格または不合格とならないパラメータ空間です。代わりに、タイミングがメタスタブルとなり読み取りが失敗する領域の間に斜めのギャップが形成される、または温度、電圧、その他の環境要因によって境界が変動する領域で構成されます。このギャップは、サンプリング クロック エッジがデータ遷移エッジを横切る遷移領域を表します。これは根本的に不安定であり、回避する必要があります。

#### 半径検証:

半径検証は、候補となるチューニング ポイントの周囲の円形領域内にあるすべてのパラメータの組み合わせをテストする検証手法です。選択されたポイントが全方向において十分なマージンを有することを保証します。

#### 対角線検索:

対角線検索は、最新のチューニング アルゴリズムで使用されるコア検索戦略です。パラメータ空間を水平方向および垂直方向に探索する代わりに、アルゴリズムは、合格領域とメタスタビリティ ギャップの両方を効果的に横断する 45 度の対角方向に沿って探索します。

### 3 PHY のチューニング アルゴリズム

PHY が有効な場合、PHY チューニングは最適なタイミング パラメータを決定する自動キャリブレーション プロセスです。

このプロセスには、以下が含まれます：

- アタック ベクタと呼ばれるテスト パターンをフラッシュ メモリに書き込みます。
- 各種タイミング パラメータの組み合わせをシステムとしてテストします。
- 各パラメータの組み合わせにおいて、テスト パターンのフラッシュ読み出しを検証します。
- フラッシュ読み取りが一貫して成功する領域を識別します。
- 最大マージンを持つ安定領域内で最適なチューニング ポイント (OTP) を選択します。

システムが高速でフラッシュ メモリに確実にアクセスできるようにするには、チューニング アルゴリズムが正常に完了する必要があります。チューニングを行わない場合、システムは性能を制限する保守的なタイミング設定を必要とするか、または温度や電圧の極端な条件においてデータ破損のリスクが高まります。

---

#### 注

アルゴリズム、ステップ、パラメータ、および対応する値は、[マイコン + SDK 11.02](#) に従って構成されています。

---

## 4 主要なチューニング パラメータ

### 4.1 非 DQS PHY チューニング アルゴリズムのパラメータ構成

パラメータ	目的	標準値	影響
最小読み取り遅延	読み取り遅延探索の下限	0	ハードウェアで定義されます
最大読み取り遅延	読み取り遅延探索の上限	3	ハードウェアで定義されます
RxDLL 探索 – TxDLL ハイエンド	RxDLL ウィンドウで探索する Tx DLL の最大値	127	RxDLL ウィンドウ探索中は TxDLL 値として固定します
RxDLL 低探索開始	Rx DLL を起動して下限境界を探索	0	ウィンドウ探索を開始するための最小 Rx DLL を定義します
RxDLL 高探索終了	より高い境界検索のために Rx DLL を終了します	127	ウィンドウ終了を検索する最大 Rx DLL を定義します
RxDLL と TxDLL の探索ステップ	探索時に Rx と Tx DLL を増減するステップ サイズ	8	値が小さいほど分解能は高くなりますが、チューニング時間は長くなります

### 4.2 DQS PHY チューニング アルゴリズムのパラメータ構成

パラメータ	目的	標準値	影響
探索半径	候補チューニング ポイントの周囲で検証する円形領域の大きさ	10	値が大きいくほどマージンは増加しますが、チューニング時間も増加します
最小パス サイズ	合格領域が有効と見なされるために必要な最小二乗長です	100	値が大きいくほど、十分なマージンを欠く狭い領域が除外されます
連続パス ポイント	タスタビリティ ギャップを確認するために必要な隣接する合格ポイントの数	10	値を大きくすると安定性検出が向上しますが、検証が困難な場合があります
連続失敗ポイント	メタスタビリティ ギャップを確認するために必要な隣接する不合格ポイントの数	5	孤立した不合格と実際のギャップを区別するのに役立ちます
対角線シフト	メインの対角線探索が失敗した場合に探索対角線をどれだけシフトするか	10	値が大きいくほど探索は高速化しますが、網羅性は低下します
最大対角線シフト	障害を宣言する前の対角線シフトの上限	70	パラメータ空間をどの程度網羅的に探索するかを決定します
読み取り遅延探索ステップ	有効な読み出し遅延値を最初に探索するときのステップサイズ	16	値が大きいくほど探索は早く完了しますが、狭い合格領域を見逃す可能性があります
DLL の最小値	DLL パラメータ探索の下限	0	ハードウェアで定義されます
DLL の最大値	DLL パラメータ探索の上限	127	ハードウェアで定義されます
最小読み取り遅延	読み取り遅延探索の下限	0	ハードウェアで定義されます
最大読み取り遅延	読み取り遅延探索の上限	4	ハードウェアで定義されます

## 5 PHY チューニング アルゴリズムの前提条件

### 5.1 ハードウェア要件

#### 5.1.1 フラッシュ デバイスの準備

- フラッシュ デバイスが正しく初期化され、アクセス可能であることを確認します。
- デバイスが目的のプロトコル モードをサポートしている必要があります。
- アタック ベクタを書き込む領域からフラッシュを消去します。

#### 5.1.2 PHY の構成

- リファレンス クロックは安定し、正しい周波数で動作している必要があります。
- 動作周波数に応じて PHY モード (マスタまたはバイパス) を設定します。
- DLL ロック モード (フル サイクルまたはハーフ サイクル) を適切に構成します。

### 5.2 アタック ベクタ

アタック ベクタは、タイミング違反を確実に検出するように設計された特定の 128 バイト パターンです。

パターンには次のものが含まれます:

- 頻繁なデータ遷移を発生させる交互ビット パターン。
- オフセットの問題を検証するための 0 および 1 の長い連続パターン。
- ビット間絶縁をテストするためのシングル ビット遷移。
- DQS モードにおける両方のサンプリング位相をカバーするための偶数および奇数バイトのアライメント。

調整が開始する前に、既知のフラッシュ アドレスにアタック ベクタを書き込みます。このアドレスを保持:

- フラッシュのアクセス可能で保護されていない領域に配置します。
- ブートローダーやアプリケーション コードに干渉しない場所。
- パターンを書き込む前に適切に消去します。

チューニング中、アルゴリズムはこのアドレスから繰り返し読み取りを行い、その結果を期待されるパターンと比較します。不一致が発生した場合、パラメータの組み合わせにおいてタイミング違反があることを示します。

### 5.3 合格領域と不合格領域の比較

半周期ロックを用いた DQS PHY チューニング アルゴリズムの遅延値は、次の次元を持つ三次元配列に格納されます:

- 第一次元: 読み取り遅延値 [0 ~ 4] – 合計 5 の値。
- 第二次元: 送信 DLL 値 [0 ~ 127] – 合計 128 の値。
- 第三次元: 受信 DLL 値 [0 ~ 127] – 合計 128 の値。

これにより、サイズ [5][128][128] の配列が生成され、合計の 81,920 要素を含みます。配列は、array[readDelay][TxDLL][RxDLL] としてインデックス指定されます。

各配列要素には、単純なバイナリ結果が格納されます:

- 通過領域は値でマークされます: 1 (攻撃ベクトルの読み取りが成功したことを示します)。
- 不具合のある領域は、次の値でマークされます: 0 (読み取りエラーまたはデータの不一致を示します)。

例:

- array[2][64][64] = 1: 読み取り遅延 = 2, Tx DLL = 64, Rx DLL = 64 の場合、テストに合格しました。
- array[2][65][65] = 0: 読み取り遅延 = 2, Tx DLL = 65, Rx DLL = 65 のとき、テストは失敗しました。

## 5.4 マスタ モードとバイパス モードの比較

OSPI PHY は、DLL 遅延値の解釈および適用方法を変える、根本的に異なる 2 つのモードで動作できます。

**マイコン + SDK 11.02** は、設定されたリファレンス クロック周波数に基づいて、適切なモードを自動的に選択します：

- リファレンス クロック < 166MHz: バイパス モードを選択します。
- リファレンス クロック  $\geq$  166MHz: マスタ モードが選択されています。

### 5.4.1 バイパス モード

バイパスモードでは、DLL 値が使用される物理的な遅延素子の数を直接制御します。各遅延素子は固定バッファまたはゲートであり、小さい絶対伝搬遅延を追加します。バイパス モードでは、Tx DLL と送信クロック信号を遅延させるために使用されるハードウェア素子の数との間に直接的な対応関係があります。合計遅延は、個々の要素遅延の合計になります。このモードの遅延時間は、動作クロック周波数にかかわらず固定時間です。

### 5.4.2 マスタ モード

マスタ モードでは、マスタ DLL 回路が遅延チェーンをロックすることで、1 クロック サイクルまたは半クロック サイクルの遅延を生成しながら、クロック周期を継続的に測定します。送信および受信の DLL 値は、この基準遅延に対する比率として解釈され、絶対的な遅延素子数としてではありません。**マイコン + SDK 11.02** では、以下の 2 種類のクロック供給メカニズムがサポートされています：

- フル サイクル ロック: マスタ DLL がフル クロック サイクルにロックする場合、DLL 値 64 は 1 クロック周期の  $64/128 = 50\%$  を表します。
- 半周期ロック: マスタ DLL が半クロック サイクルにロックする場合、DLL 値 64 は  $64/128 =$  半周期の 50% を表し、これは 1 クロック周期の 25% に相当します。

マスタ モードでは、遅延はクロック周期に比例し、周波数に応じて自動的にスケーリングされます。半周期ロックは、クロック サイクルのより小さな区間において高精度なタイミング制御を実現するためのより高い分解能を提供します。このモードでは、遅延はクロック周期に対する比率として指定され、絶対的なクロック周波数に依存せず適切なタイミング関係を維持します。これは、高速動作において、セットアップ時間およびホールド時間をクロック エッジに対して厳密に制御する必要がありますため、信頼性の高い動作に不可欠です。

## 6 新しいチューニング アルゴリズムの必要性

新しい DQS アルゴリズムは、従来のアルゴリズムにおける信頼性の重大な制限を解決します。来の手法では、領域サイズや温度依存の配置に主に基づいてチューニング ポイントを選択しており、選択されたポイント周辺のマージンを検証していなかったため、極端な位置に近いチューニング ポイントとなり、ノイズに起因する変動の影響を受けやすくなっていました。従来の手法では、固定された Tx/Rx 探索によるコーナー検出および境界に沿った二分探索を用いており、狭い領域を見逃したり、不規則な合格領域の形状を誤って評価したりする可能性があります。さらに、従来の手法では、主探索が失敗した場合のフォールバック戦略が不足していました。新しい PHY チューニング アルゴリズムは、全方向に十分なマージンを確保するための円形半径検証、連続する合格および不合格ポイント数を用いた明示的なメタスタビリティ ギャップ検出 (孤立したノイズを除外)、DQS の合格領域の自然な幾何形状に従う対角探索、最適配置のための幾何学的中点計算、最小領域サイズの強制、およびフォールバック戦略としての対角シフトを導入しています。これらの強化により、多様な基板設計、フラッシュ デバイス、および PVT 条件にわたって信頼性の高い動作を実現できます。

### 6.1 温度変化

温度は半導体のタイミングに大きな影響を及ぼします：

- 温度が高くなると動作が遅くなります。温度が上昇すると、シリコン結晶格子内の原子振動が増加するため、電子はトランジスタ内をより遅く移動します。
- 温度が低いほど動作は高速になります。低温では原子格子の振動が抑えられるため、電子はトランジスタ内をより自由かつ高速に移動できます。
- 一般的な動作温度範囲 ( $-40^{\circ}\text{C} \sim +125^{\circ}\text{C}$ ) において、遅延素子は伝搬遅延特性を変化させる可能性があり、その結果、パラメータ空間における DLL の位置がシフトします。

メタスタビリティ ギャップは、温度の変化に伴って信号間の相対的なタイミングが変動するため、位置が移動します。新しい DQS チューニング アルゴリズムは、十分なマージンを持つポイントを選択することで補償し、合格領域が温度範囲全体にわたって有効であり続けるようにします。

古いアルゴリズムと新しいアルゴリズムによって計算された OTP 値の比較を以下の図に示します。

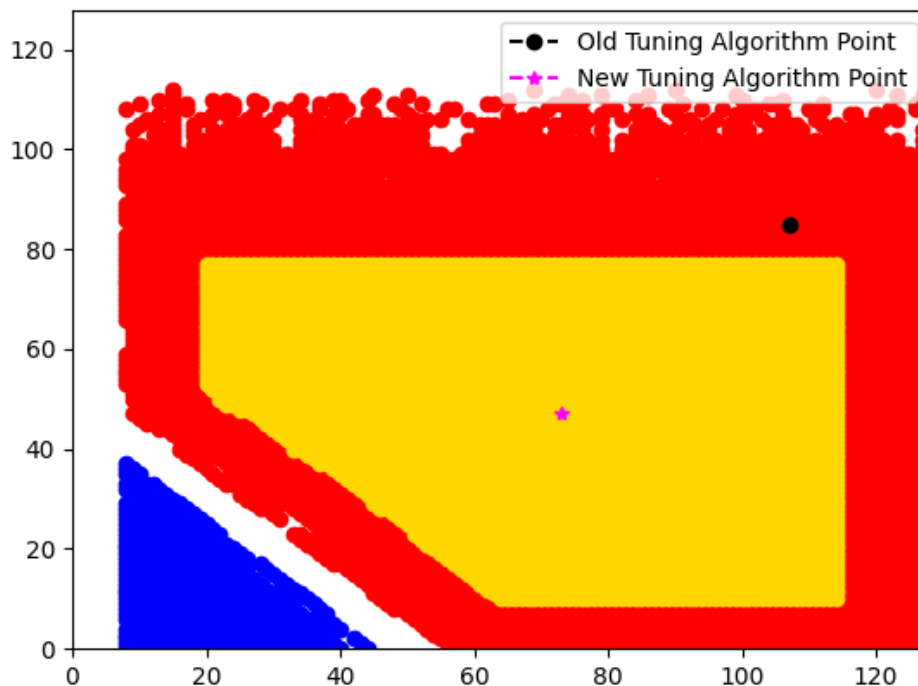


図 6-1. 新旧の OTP の比較

## 7 アルゴリズムの実装

### 7.1 DQS PHY のチューニング アルゴリズム

DQS チューニング アルゴリズムは、DQS シグナリングを用いた高速動作において最適なタイミング パラメータを特定するために、対角線探索戦略を使用します。

#### 7.1.1 対角線の選択

アルゴリズムは、(0, 0) から (127, 127) までの対角線に沿って検索を開始し、TX および RX DLL の値が等しい増分を表します。有効なチューニング ポイントが見つからない場合、対角線は 10 ポイント上方向 (Y オフセットを増加) または 10 ポイント右方向 (X オフセットを増加) に、交互のパターンでシフトされます。シフトは元の対角線から最大 70 ユニットまで継続するため、45 度の傾斜を維持しながらパラメータ空間を体系的にカバーできます。

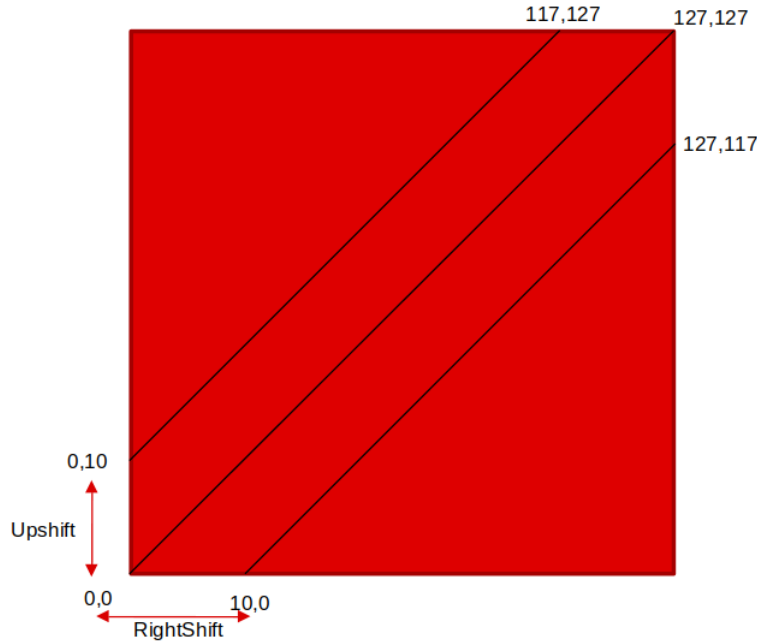


図 7-1. 対角線の選択

#### 7.1.2 有効な読み取り遅延の選択

最小の読み取り遅延から開始し、アルゴリズムは対角線に沿って 16 ステップ刻みの粗い探索を行い、読み取りが成功する領域を特定します。合格ポイントは、読み取り遅延設定において有効な動作領域が存在することを示します。一度検出されると、1 つのマップのステップで詳細な検索が有効な領域全体になります。このプロセスは、読み取り遅延値ごとに繰り返されます。すべての読み取り遅延で通過点が見つからない場合は、対角線をシフトして新しい潜在的な動作点を見つけます。

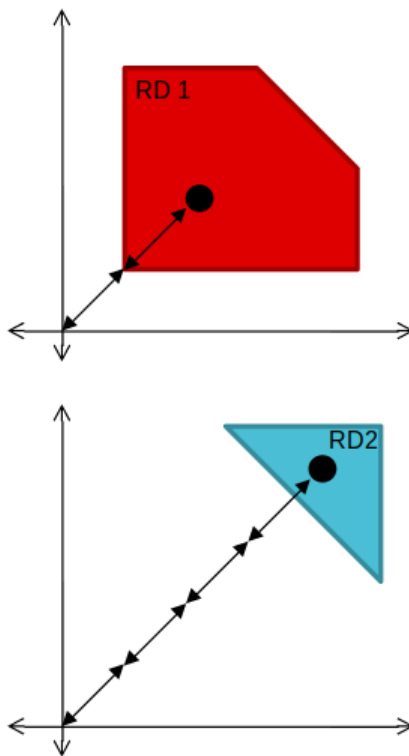


図 7-2. 有効な読み取り遅延の選択

### 7.1.3 コーナーポイントの識別

アルゴリズムは、対角線に沿った合格領域の正確な境界を特定します。最小の有効な読み取り遅延に対して、アルゴリズムは対角線の開始点から Tx DLL および Rx DLL を 1 ステップずつインクリメントし、最初の合格ポイント (下限境界) を見つけるまで探索します。最大の有効な読み取り遅延に対して、アルゴリズムは対角線の終点から両方の DLL をデクリメントし、最初の合格ポイント (上限境界) を見つけるまで探索します。これらのコーナーポイント、インターフェイスが確実に動作する範囲全体を定義します。

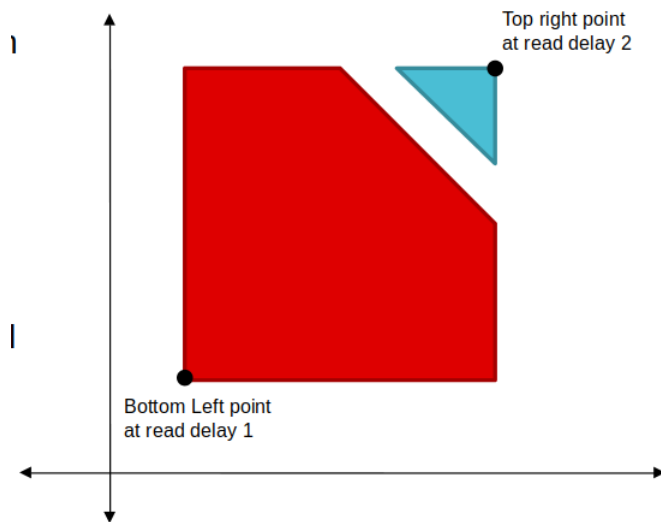


図 7-3. コーナーポイントの識別

### 7.1.3.1 1つの読み取り遅延値に対してのみコーナーポイントを選択

両方のエンドポイントに同一の読み取り遅延が存在する場合、単一の通過領域が存在します。アルゴリズムは、連続する合格ポイント数が最も多いラインのコーナーポイントを特定し、安定したサンプリングウィンドウを表します。コーナーポイント間の距離であるライン長が最小合格長スレッシュホールドを超える場合、チューニングポイントの選択が進行します。それ以外の場合、ウィンドウが信頼性の高い動作には狭すぎるため、失敗が報告されます。

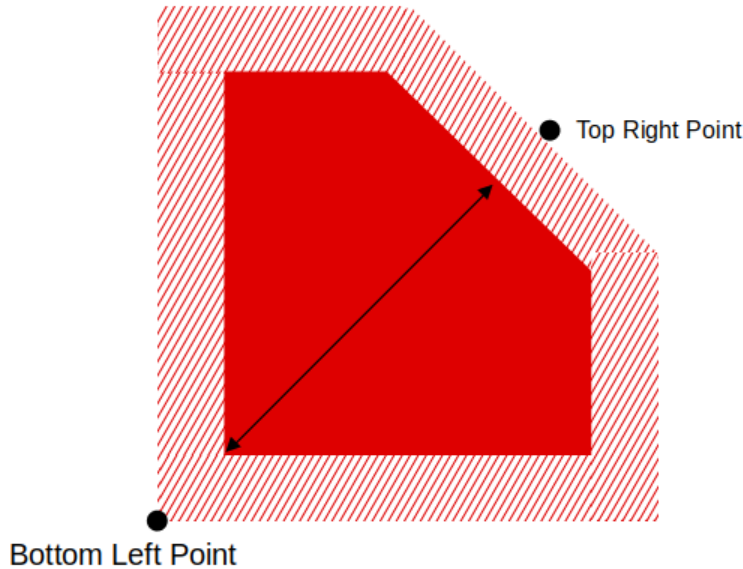


図 7-4. 1つの読み取り遅延値に対するコーナーポイントの選択

### 7.1.3.2 2つの異なる読み取り遅延値に対するコーナーポイント選択

対角線の端点で異なる読み取り遅延が表示される場合、メタスタビリティギャップによって分離された2つの合格領域が存在します。アルゴリズムは、連続する最小の合格ポイント数を満たすポイントを特定し、複数の隣接ポイントをテストすることで安定性を向上させながらコーナーポイントを精緻化します。各有効ポイントについて、アルゴリズムは最初の不合格ポイントを特定し、明確な境界を確立するために最小連続不合格ポイント数を確認します。連続通過点が十分でない地域は不安定な状態で排除されます。アルゴリズムは、対角線に沿って上方向に最小の読み取り遅延値を探索し、下方向に最大の読み取り遅延値を探索します。

精緻化の後、アルゴリズムは合格ポイント数が最も多いラインを特定します。いずれかのライン長が最小合格長を超える場合、チューニングポイントの選択が進行し、それ以外の場合はキャリブレーション失敗が報告されます。

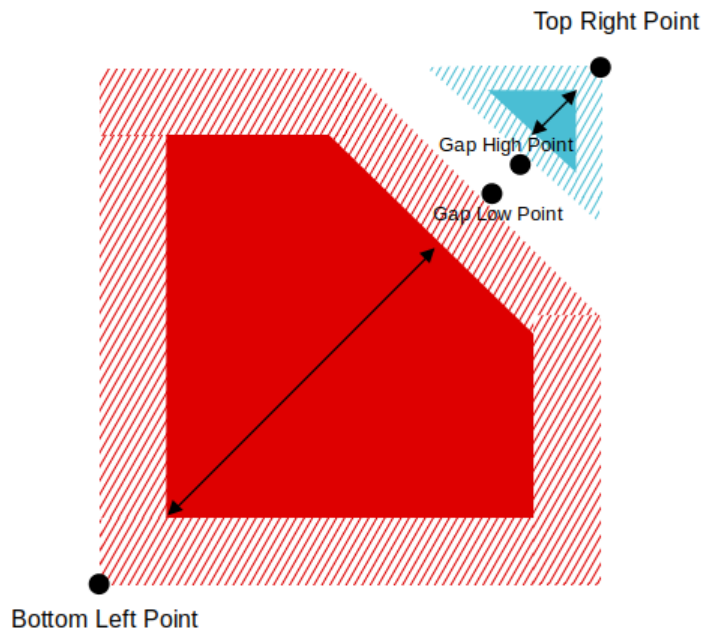


図 7-5. 2つの異なる読み取り遅延値に対するコーナーポイント選択

#### 7.1.4 チューニングポイントの選択

このアルゴリズムは、より長い長さの領域を選択し、中間点 (midpoint1) を計算します。midpoint1 における対角線に垂直な直線のコーナーポイントが特定され、連続して合格するポイントを表します。垂直線の中間点 (midpoint2) が計算され、安定領域の中心幅を表します。半径検証では、midpoint2 を中心とした指定半径の円内にあるすべてのポイントが合格するかどうかを検証します。成功すると、midpoint2 が最適なチューニングポイントになります。

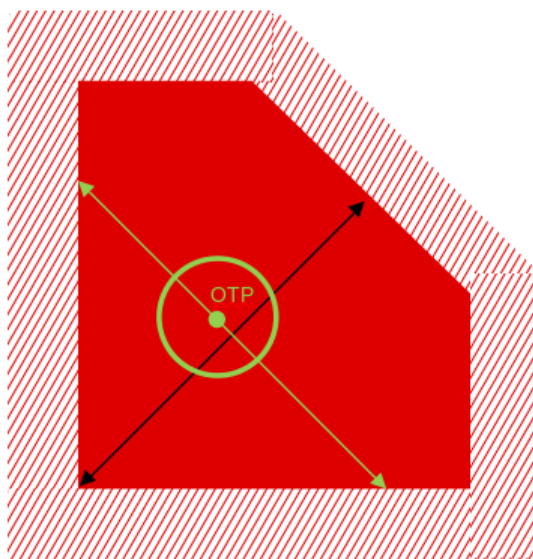


図 7-6. 中間点でのチューニングポイントの選択

半径検証が失敗した場合、垂直線は midpoint1 において 2 つのセグメントに分割されます。大きなセグメントの midpoint (midpoint3) で半径の検証が行われます。成功すると、midpoint3 がチューニングポイントになります。

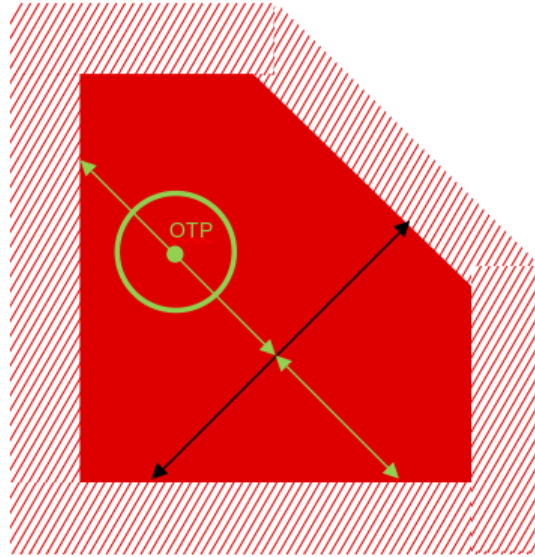


図 7-7. 大きなセグメントの中間点でのチューニングポイントの選択

midpoint2 と midpoint3 の両方が不合格の場合、長さが最小合格長を超えていれば二番目の読み取り遅延領域が評価され、選択プロセスが繰り返されます。

すべての領域およびセグメントが不合格の場合、アルゴリズムはチューニング失敗を報告し、網羅的なカバレッジを実現するために定義済みパターンに従って別の対角線を選択します。各対角線は異なる遅延パラメータの組み合わせを表し、

安定した動作領域を見つけるための複数の機会を提供します。

PHY 障害の場合、[一般的な起動の問題とデバッグアプリケーション ノートの「PHY 故障」セクション](#)を参照してください。

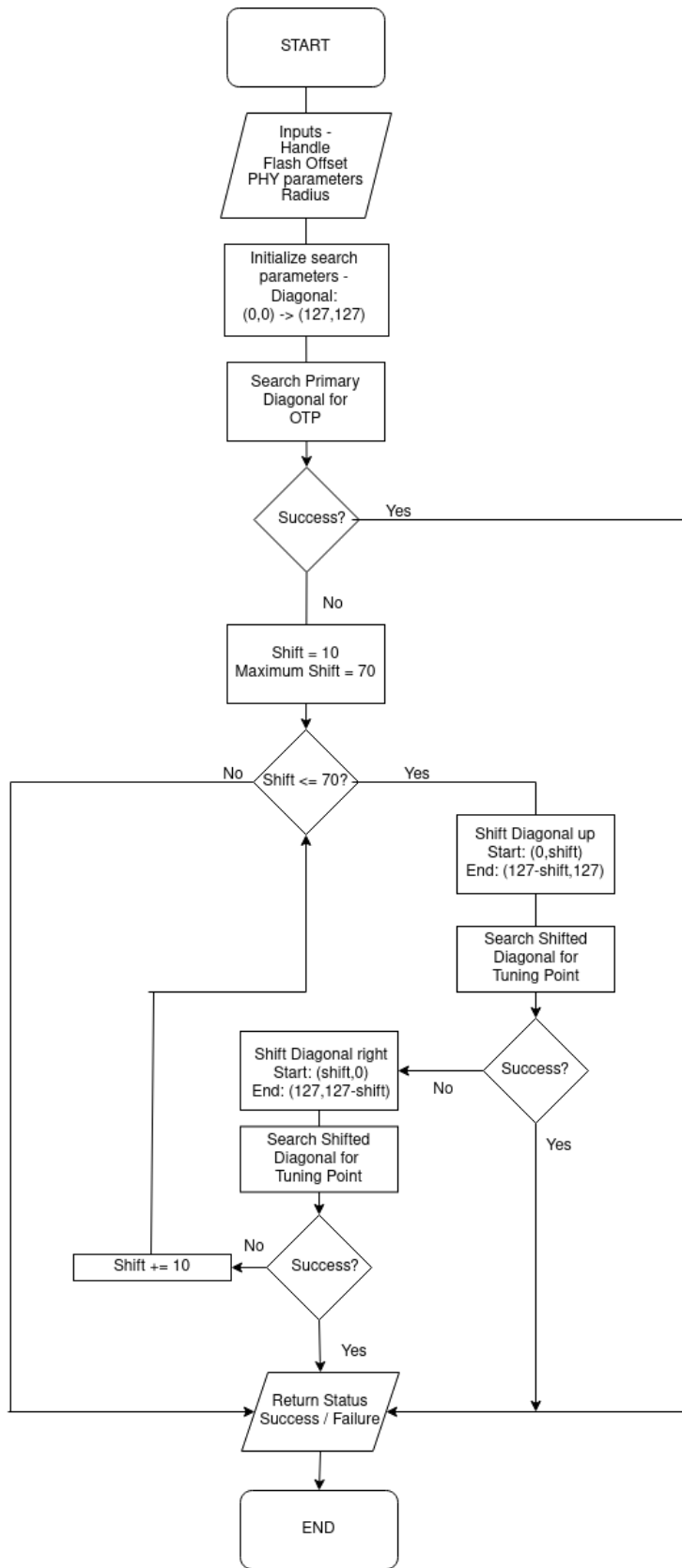


図 7-8. DQS チューニング アルゴリズムのフローチャート

## 7.2 非 DQS PHY チューニング アルゴリズム

非 DQS チューニング アルゴリズムは、DQS 信号サポートのない低速動作用に設計された、簡略化されたウィンドウ ベースのアプローチを使用します。DQS アルゴリズムにおける二次元の対角探索とは異なり、非 DQS アルゴリズムは Tx DLL 値を固定したまま Rx DLL 軸に沿った一次元探索を行います。

### 7.2.1 Tx DLL の値を固定

アルゴリズムは、Tx DLL をあらかじめ設定された高い値に設定することから開始します。この固定構成により、非 DQS モードをサポートするほとんどのフラッシュ デバイスに対して十分なセットアップ時間が確保され、探索問題が二次元から一次元に削減されます。Tx DLL を最適化する必要がなくなるので、このアルゴリズムは最適な Rx DLL タイミングを見つけることだけに集中できます。

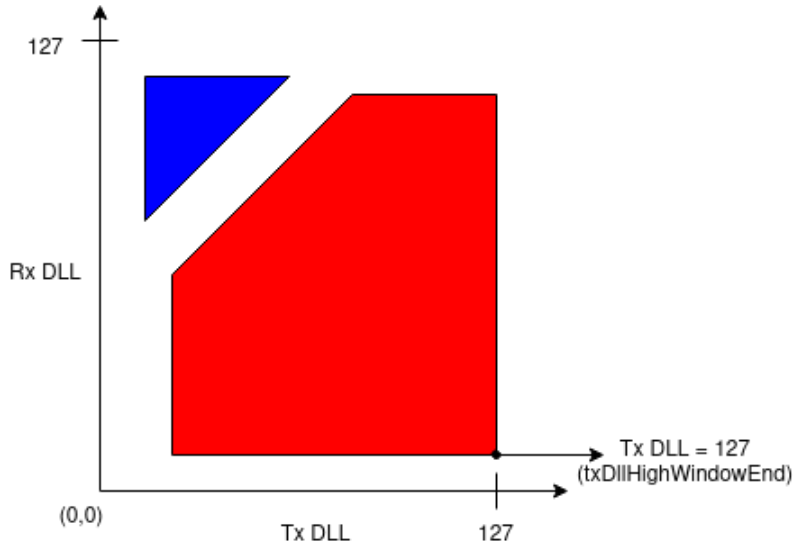


図 7-9. Tx DLL の値を固定

### 7.2.2 Rx ウィンドウ 1 を検出

最小読み取り遅延値から開始し、アルゴリズムは最初の有効な Rx DLL ウィンドウを探索します。各ステップでアタック ベクトルの読み取りをテストしながら、Rx DLL は最小値から増加します。読み取りが成功した最初の DLL 値はウィンドウ開始 (rxStart1) になり、失敗する前の最後の値はウィンドウ終了 (rxEnd1) になります。ウィンドウ サイズは、 $rxWindow1 = rxEnd1 - rxStart1$  として計算されます。

現在の読み取り遅延で合格ウィンドウが見つからない場合、アルゴリズムは読み取り遅延を増加させて探索を繰り返します。このプロセスは、ウィンドウが検出されるか、最大読み取り遅延を超えるまで継続されます。合格ウィンドウは、インターフェイスのタイミングがすべてのセットアップ時間およびホールド時間の要件を満たす、連続した受信 DLL 値の範囲を表します。

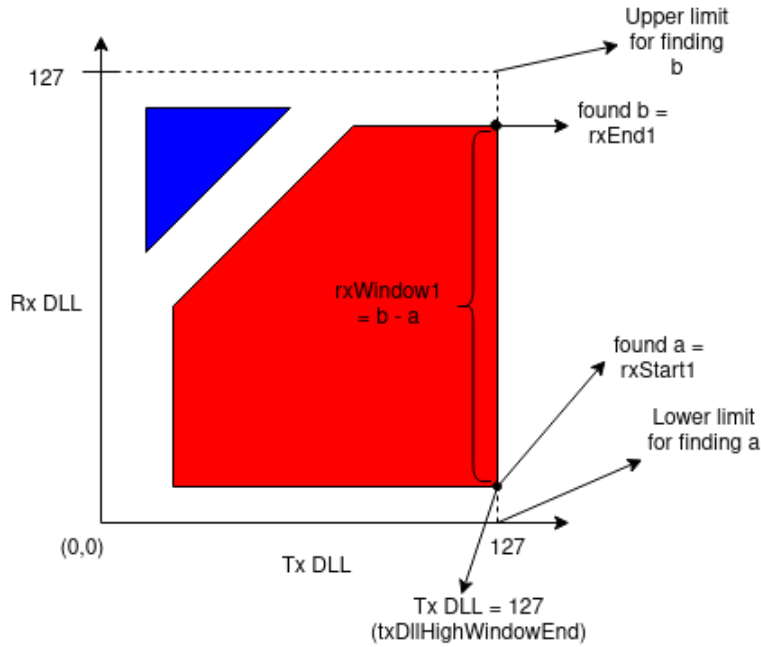


図 7-10. Rx ウィンドウ 1 の選択

### 7.2.3 Rx ウィンドウ 2 を検出

最初のウィンドウを特定した後、アルゴリズムは次に高い読み取り遅延値 (ウィンドウ 1 の読み取り遅延 + 1) で 2 番目のウィンドウを特定しようとします。rxStart2 と rxEnd2 を検索するために同じ検索プロセスを繰り返し、2 番目のウィンドウサイズは  $rxWindow2 = rxEnd2 - rxStart2$  として計算されます。

2 番目のウィンドウが見つからない場合、rxWindow2 は 0 に設定されます。これは失敗とはみなされません。読み取り遅延値が異なると、リファレンスクロックとデータサンプリングポイント間の相対タイミングがシフトします。複数の読み取り遅延値をテストして、アルゴリズムが使用可能な最大ウィンドウを検出し、環境の変動に対して最大マージンを提供するようにします。

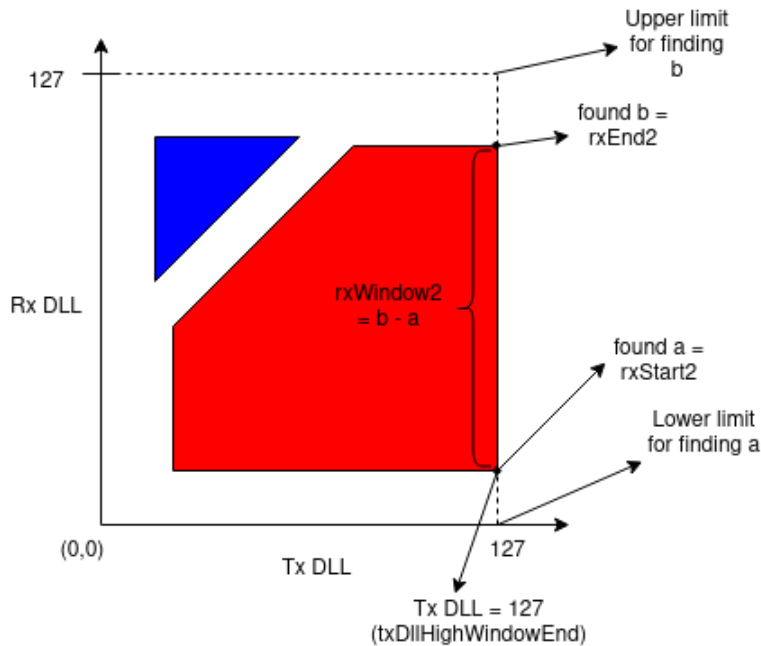


図 7-11. Rx ウィンドウ 2 の選択

### 7.2.4 より大きい Rx ウィンドウを選択

アルゴリズムは 2 つのウィンドウ サイズを比較し、大きい方を選択します。rxWindow2 が rxWindow1 より大きい場合、対応するパラメータ (rxStart2、rxEnd2、および読み取り遅延) とともにウィンドウ 2 が選択されます。それ以外の場合は、ウィンドウ 1 が使用されます。より大きなウィンドウは、より大きなタイミング マージンと、温度変化、電圧変動、および製造のばらつきに対するより高い耐性を示します。

### 7.2.5 OTP の計算

最終的なチューニング ポイントは、次の三つのパラメータを使って決定されます：

1. 送信 DLL: ステップ 1 からの固定値。
2. 読み取り遅延: 選択したウィンドウに関連付けられた読み取り遅延。
3. 受信 DLL: 選択したウィンドウの中心 =  $rxStart + (rxEnd - rxStart) / 2$ 。

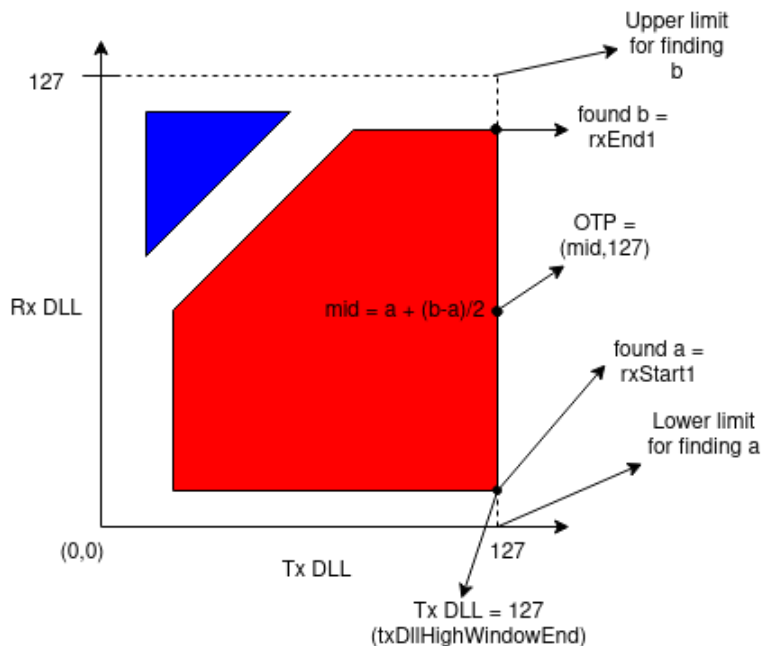


図 7-12. OTP の計算

### 7.2.6 温度に関する検討事項

SoC の内部温度センサである VTM モジュールは、現在のダイ温度を測定します。基準温度 42.5°C を用いて、アルゴリズムは温度依存のオフセットを計算します：

$$\text{温度係数} = (\text{現在の温度} - 42.5^\circ\text{C}) / 165^\circ\text{C} \times \text{ウィンドウ サイズ} \times 0.75$$

調整後の受信 DLL は、中間点と温度係数の差と等しくなります。温度が高くなると遅延が増加するため、チューニング ポイントはより低い受信 DLL 値側へシフトします。温度が低いと、遅延が減少するため、チューニング ポイントはより高い値にシフトします。

チューニング ポイントを計算した後、これらのパラメータは PHY のハードウェア レジスタに適用されます。アタック ベクタの検証読み出しにより、アルゴリズムが成功を返す前にチューニングの成功が確認されます。

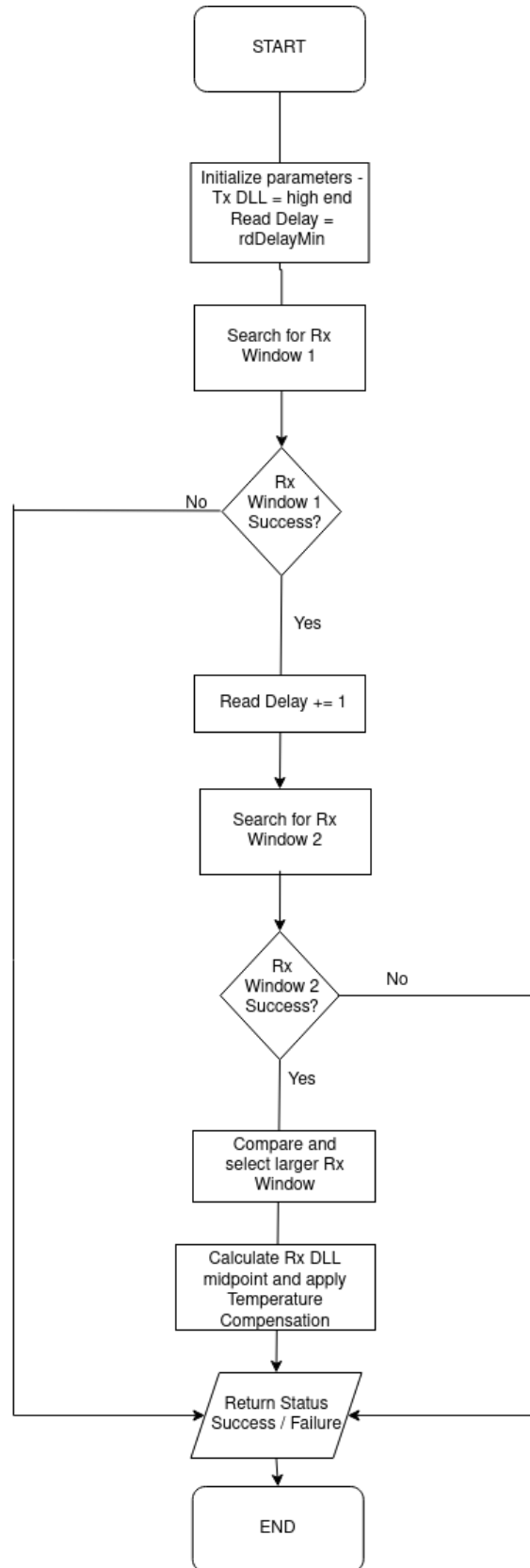


図 7-13. 非 DQS チューニング アルゴリズムのフローチャート

## 8 チューニングの拡張

### 8.1 調整時間最適化 – スキップ調整機能

**マイコン + SDK 11.02** には、PHY チューニングをスキップする機能があります。この機能により、後のステージのブート時間が短縮されます。マルチステージブートプロセスを持つシステムでは、プライマリブートローダーなどの最初のステージでチューニングを一回実行し、次の手順を実行することで後続のステージで再利用できます：

1. 最初のブートステージは、完全なチューニングを実行し、パラメータ値をハードウェアレジスタに保存します。
2. **SysConfig** でスキップ チューニングが有効になっている場合、後続のステージは保存されたパラメータを取得し、ドライバによって直接適用されます。

### 8.2 ランタイム検証 – OTP の検証

**マイコン + SDK 11.02** には、OTP を検証する機能があります。**SysConfig** で有効にすると、アプリケーションは PHY 有効時のフラッシュ読み取り動作中にチューニング ポイントを再検証します。検証プロセスは、次の手順を実行することで動作します：

1. フラッシュ読み取り動作中、**SysConfig** で有効になっている場合、OTP 検証が自動的に実行されます。
2. この機能は、設定された半径の円形領域内の現在のチューニング ポイントの周りで対角線チェックを実行します。
3. 検証に失敗した場合、PHY チューニング アルゴリズムは自動的に再実行されます。

## 9 まとめ

OSPI PHY チューニング アルゴリズムは、信頼性の高い高速フラッシュ メモリ アクセスを実現するためにタイミング パラメータを自動的にキャリブレーションします。DQS モードでは、対角探索戦略を用いて合格領域を効率的に特性評価し、最適な中点を算出し、円形領域テストによりマージンを検証します。非 DQS モードでは、温度補償を伴うより単純な一次元ウィンドウ探索を使用します。最新の DQS チューニング アルゴリズムは、半導体タイミング特性に影響する製造上のばらつきや環境変化を補正します。エンジニアは、スキップ チューニングを使用してブートを高速化するか、本番環境では事前定義を使用して最適化を行うことができます。

## 10 参考資料

1. テキサス インストルメンツ、[AM62x マイコン + SDK](#)。
2. テキサス インストルメンツ、[OSPI コントローラ PHY チューニング アルゴリズム](#)、アプリケーション ノート
3. テキサス インストルメンツ、[OSPI PHY チューニング アルゴリズム](#)、Github リポジトリ
4. テキサス インストルメンツ、[マイコン + SDK における OSPI および QSPI シリアル NOR/NAND フラッシュ動作のルーブット特性評価](#)、アプリケーション ノート
5. テキサス インストルメンツ、[マイコン + SDK 用 xSPI カスタム フラッシュ デバッグ ガイド](#)、アプリケーション ノート

## 重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとし、

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、[TI の販売条件](#)、[TI の総合的な品質ガイドライン](#)、[ti.com](#) または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

Copyright © 2026, Texas Instruments Incorporated

最終更新日 : 2025 年 10 月