

TRAVEO™ T2G ファミリの SMIF の使用方法

本書について

適用範囲と目的

本アプリケーションノートでは、TRAVEO™ T2G ファミリ MCU のシリアルメモリインタフェース (SMIF) の使用方法について説明します。

関連製品ファミリ

TRAVEO™ T2G ファミリ

対象者

このドキュメントは、TRAVEO™ T2G ファミリの SMIF を使用するユーザーを対象とします。

目次

本書について.....	1
目次	1
1 はじめに	3
2 概要.....	4
2.1 特長.....	4
2.2 ブロックダイアグラム.....	5
2.3 TX コマンド FIFO のコマンド	7
2.4 データキャプチャ方式.....	10
2.4.1 出力/フィードバッククロックベースキャプチャ	10
2.4.2 内部クロックベースキャプチャ.....	10
2.4.3 RWDS ベースキャプチャ.....	10
2.4.4 遅延ラインおよびデータ学習パターン (DLP) ベースキャプチャ	11
3 デュアルクアッド SPI DDR モードの例	12
3.1 メモリデバイスの接続.....	12
3.2 デュアルクアッドモードでの S25FL256 デバイスの使用	12
3.3 初期化.....	15
3.3.1 SMIF の初期化	16
3.3.2 クアッドのリセット.....	17
3.3.3 クアッドのレジスタ読出し.....	19
3.3.4 クアッドのレジスタ設定.....	21
3.4 消去.....	22
3.5 MMIO モードでの書込み転送	22
3.6 XIP モードでの読出し転送	24
3.7 データ学習パターン	26
3.7.1 DLP の設定.....	26
3.7.2 DLP キャリブレーション	26
4 ハードウェアコンフィギュレーション	29
4.1 デュアルクアッドコンフィギュレーションでの S25FL256 デバイスの使用	29
4.2 HYPERFLASH™ および HYPERRAM™ コンフィギュレーションの使用	30
5 用語集	31



目次

関連資料32

改訂履歴33

免責事項34

はじめに

1 はじめに

このアプリケーションノートでは、TRAVEO™ T2G ファミリ MCU と SPI デバイスへの少ないピン数での接続方法を提供します。SMIF はマスタです。このアプリケーションノートでは、SPI フラッシュメモリ S25FL256S を使用し、デュアルクアッド SPI モードでの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) の使用例を説明します。シングル/デュアル/クアッド/オクタール SPI プロトコルの詳細については、[architecture technical reference manual \(TRM\)](#) をご覧ください。SMIF 機能がサポートされているかどうかを確認するには、デバイス固有のデータシートを参照してください。

このアプリケーションノートで使用される用語については、[用語集](#)を参照してください。

概要

2 概要

2.1 特長

以下に SMIF の特長を示します。

- SPI または HYPERBUS™ マスタ機能
- HYPERBUS™ プロトコル
 - HYPERFLASH™
 - HYPERRAM™
- SPI プロトコル
 - SPI モード 0 のみ、設定可能な MISO サンプルングタイミング
 - シングル, デュアル, クアッド, およびオクタル SPI をサポート
 - デュアルクアッド SPI モードをサポート
 - 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) 転送をサポート
- メモリデバイス
 - 64 KB から 4 GB の範囲で 2 の倍数の累乗でデバイス全体の容量をサポート
 - 設定可能な外部デバイス容量をサポート
 - 2 つの外部メモリデバイスをサポート
- メモリマップド I/O (MMIO) 操作モード
- コードの直接実行 (XIP) モード
 - 読出しと書込みアクセスの両方の XIP 操作モード
 - XIP モードはオンザフライの暗号化と復号化をサポート
 - Arm® Cortex®-M0 の AHB インタフェースおよび Cortex® M7 コアの AXI インタフェースを介した XIP 操作モード
 - 最大 4 つのトランザクションをサポート
- メモリインタフェースロジック
 - FIFO のバックプレッシャに対処するための SPI および HYPERBUS™ 転送のストールをサポート
 - 非同期 SPI/HYPERBUS™ 送受信インタフェースクロックをサポート
 - 読み書きデータストローブ (RWDS) をサポート
 - 複数のインタフェース受信クロックをサポート
 - フレキシブルな外部 SPI メモリデバイスのデータ信号接続をサポート
 - PLL/FLL からの独立した SPI インタフェース送信クロック
 - SPI インタフェースロジックはフレキシブルな外部メモリデバイスのデータ信号接続をサポート

概要

2.2 ブロックダイアグラム

SMIF により、外部デバイスへ少ないピン数で接続が可能です。Figure 1 に CYT4B シリーズの SMIF のブロックダイアグラムを示します。

Figure 1 の下部は、I/O サブシステム (IOSS) への SPI インタフェース信号接続を示します。黄色でハイライトされたセクションは、AXI スレーブインタフェースと AHB-Lite スレーブインタフェースを示します。XIP AHB-Lite インタフェースには専用のキャッシュがあります。XIP アドレス空間への AHB-Lite 転送は、キャッシュにアクセスするか、オンザフライで外部デバイスへの SPI 転送に変換されます。SMIF は TRAVEO™ T2G のアドレス 0x6000:0000 にあるアドレス空間をサポートします。XIP アドレス空間内の外部デバイスの領域はプログラム可能です。

デュアルクアッド SPI モードの場合、SMIF I/O 信号インタフェースに並列に接続されている 2 つの外部デバイスに対して同じ MMIO デバイスレジスタ値をプログラムする必要があります。

AHB-Lite は、MMIO レジスタにアクセスして MMIO アドレス空間に転送します。MMIO レジスタには、FIFO にアクセスするためのレジスタが含まれます。XIP アドレス空間は、外部デバイスへ非常に効率的な読出しおよび書込みアクセスをサポートしますが (AHB-Lite 転送から SPI 転送へのオンザフライ変換により)、MMIO アドレス空間は SPI 転送の構築に柔軟性を提供します。

SMIF には、2 つの TX FIFO と 1 つの RX FIFO があります。これらの FIFO は、CLK_mem ロジックと CLK_if_tx/CLK_if_rx メモリインタフェースロジック間の非同期クロックドメイン転送を提供します。メモリインタフェースロジックは、TX および RX FIFO を介して完全に制御されます。さらに、SMIF には RX データ MMIO FIFO があります。これは MMIO モードでのみ使用され、論理的に RX データ FIFO の拡張であり、ソフトウェアでの使いやすい RX データ処理を可能にします。

SMIF には単一の割込みラインがあります。

XIP モードでは、暗号化コンポーネントは書込みデータのオンザフライ暗号化と読出しデータのオンザフライ復号化をサポートします。オンザフライの暗号化の使用は、MMIO CTL レジスタの CRYPTO_EN ビットフィールドによって決定されます。MMIO モードでは、MMIO レジスタインタフェースを介して暗号化コンポーネントにアクセスし、オフラインの暗号化と復号化をサポートします。

概要

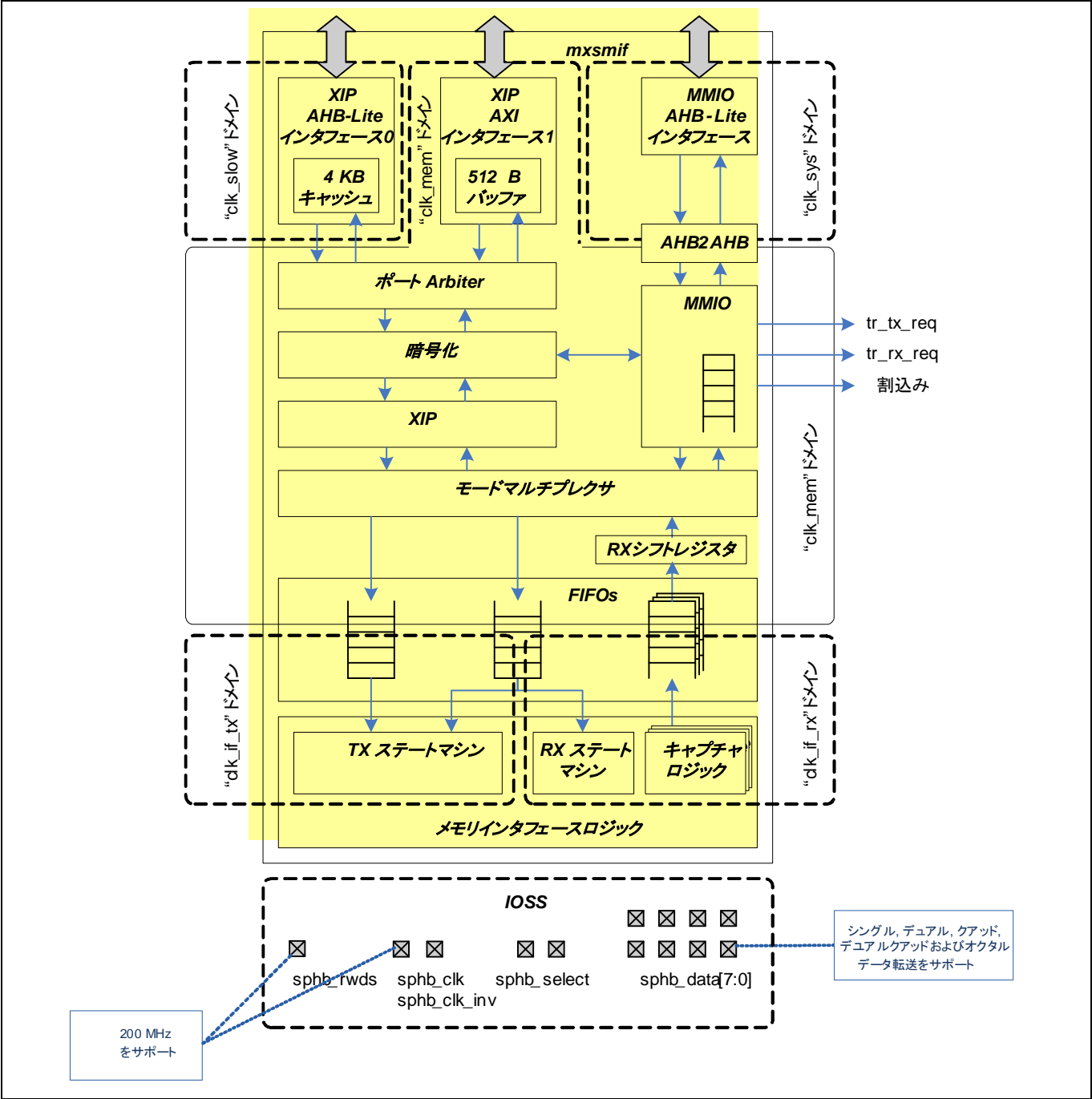


Figure 1 CYT4B シリーズのハイレベルブロックダイヤグラム

他のシリーズの割込みアーキテクチャについては、[architecture TRM](#) を参照してください。

概要

2.3 TX コマンド FIFO のコマンド

このアプリケーションノートでは、さまざまなコマンドを使用します。以下の FIFO は、SMIF でこれらのコマンドの送受信に使用されます。

- TX コマンド FIFO: クアッド SPI および HYPERBUS™のコマンドをメモリインタフェースロジックに送信します。FIFO は、8 つの 27 ビットエントリで構成されます。各エントリはコマンドを保持します。FIFO は SMIF_TX_CMD_FIFO_WR.DATA27[26:0] レジスタによって制御されます。DATA27[26:24] はコマンドを指定し、DATA[23:0] はコマンドタイプに応じてコマンド仕様を設定します。
- TX データ FIFO: 書き込みデータをメモリインタフェースロジックに送信します
- RX データ FIFO: メモリインタフェースロジックから読み出しデータを受信します

クアッド SPI のコマンドシーケンスは、フェーズごとに分類されます。フェーズには 1 バイトの命令, 4 バイトのアドレス, 1 バイトのモード, クアッドメモリで決定されたダミーサイクル, およびデータ受信があります。TX コマンド FIFO のコマンドは、シーケンスのフェーズを指定します。Table 1 に、TX コマンド FIFO がサポートする 5 種類のコマンドを説明します。コマンドの詳細については、[architecture TRM](#) および [registers TRM](#) を参照してください。

Table 1 5 種類のコマンド TX コマンド FIFO コマンド

コマンド	DATA27[26:24]	仕様
TX	0	このコマンドは、クアッド SPI のコマンドの命令, アドレス, モードなどのフェーズを指定します。
TX_COUNT	1	このコマンドは、データが TX データ FIFO から外部メモリに送信されるときに使用されます。このコマンドは、送信するメモリデータユニットの数を指定します。
RX_COUNT	2	このコマンドは、外部メモリから RX データ FIFO にデータを受信するときに使用されます。このコマンドは、受信するメモリデータユニットの数を指定します。
DUMMY_COUNT	3	このコマンドは、ダミーサイクルの数を指定します。
DESELECT	4	このコマンドにより、メモリインタフェース送信ロジックが転送を終了し、メモリデバイスの選択を解除します。

概要

Figure 2 に、クアッド SPI の読出しコマンドに TX コマンド FIFO を使用する方法を説明します。

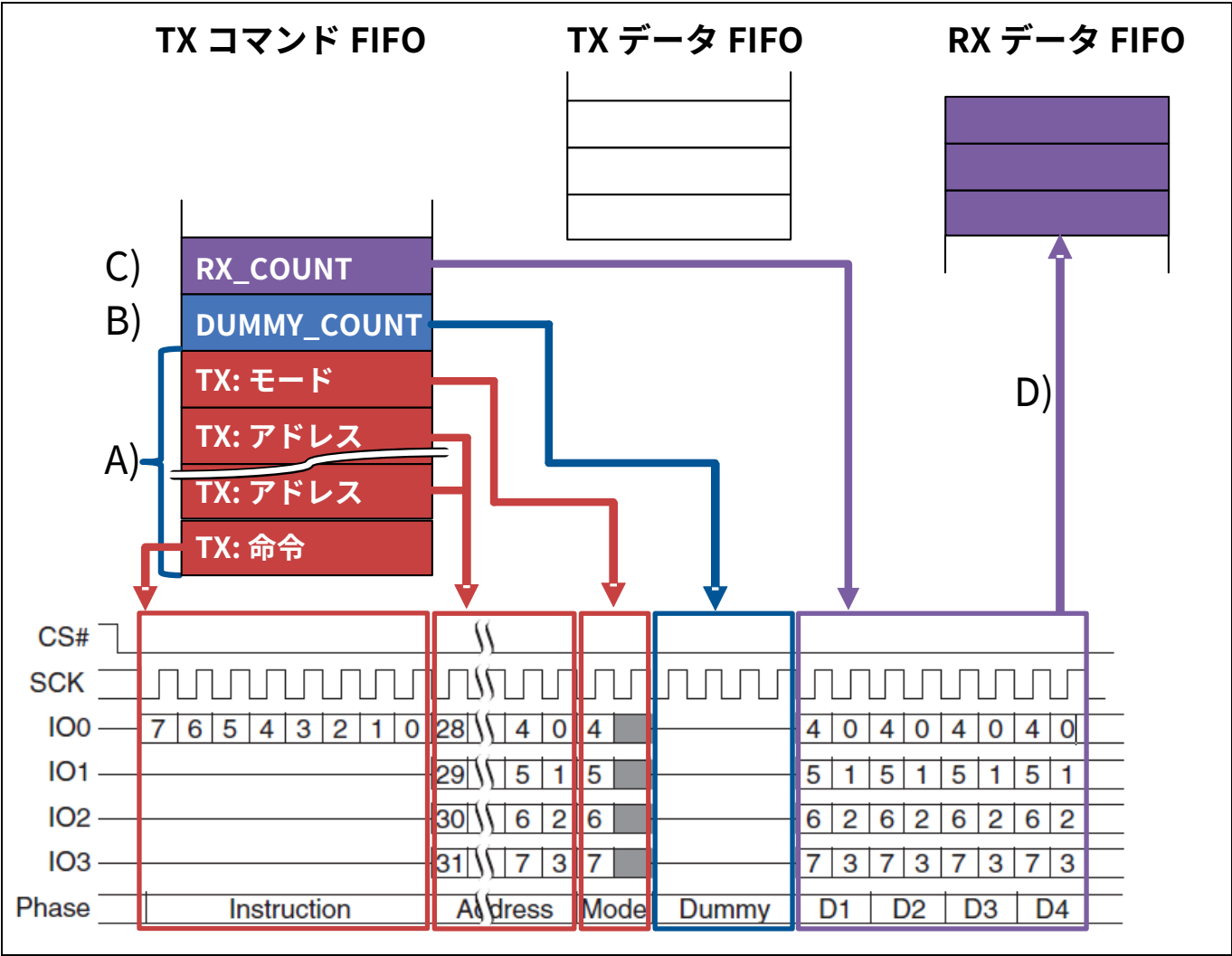


Figure 2 クアッド SPI の読出しコマンドの構築

クアッド SPI の読出しコマンドの TX コマンド FIFO の設定を以下に説明します。

- A) “TX” コマンドを使用して、命令、アドレス、モードの順に TX コマンド FIFO にエントリを設定。
- B) “DUMMY_COUNT” コマンドを使用して、TX コマンド FIFO にダミーサイクルを設定。
- C) “RX_COUNT” コマンドを使用して、TX コマンド FIFO に受信データの数を設定。
- D) RX データ FIFO はデータセットの数を受信。

概要

Figure 3 に、クアッド SPI のプログラムコマンドに TX コマンド FIFO を使用する方法を説明します。

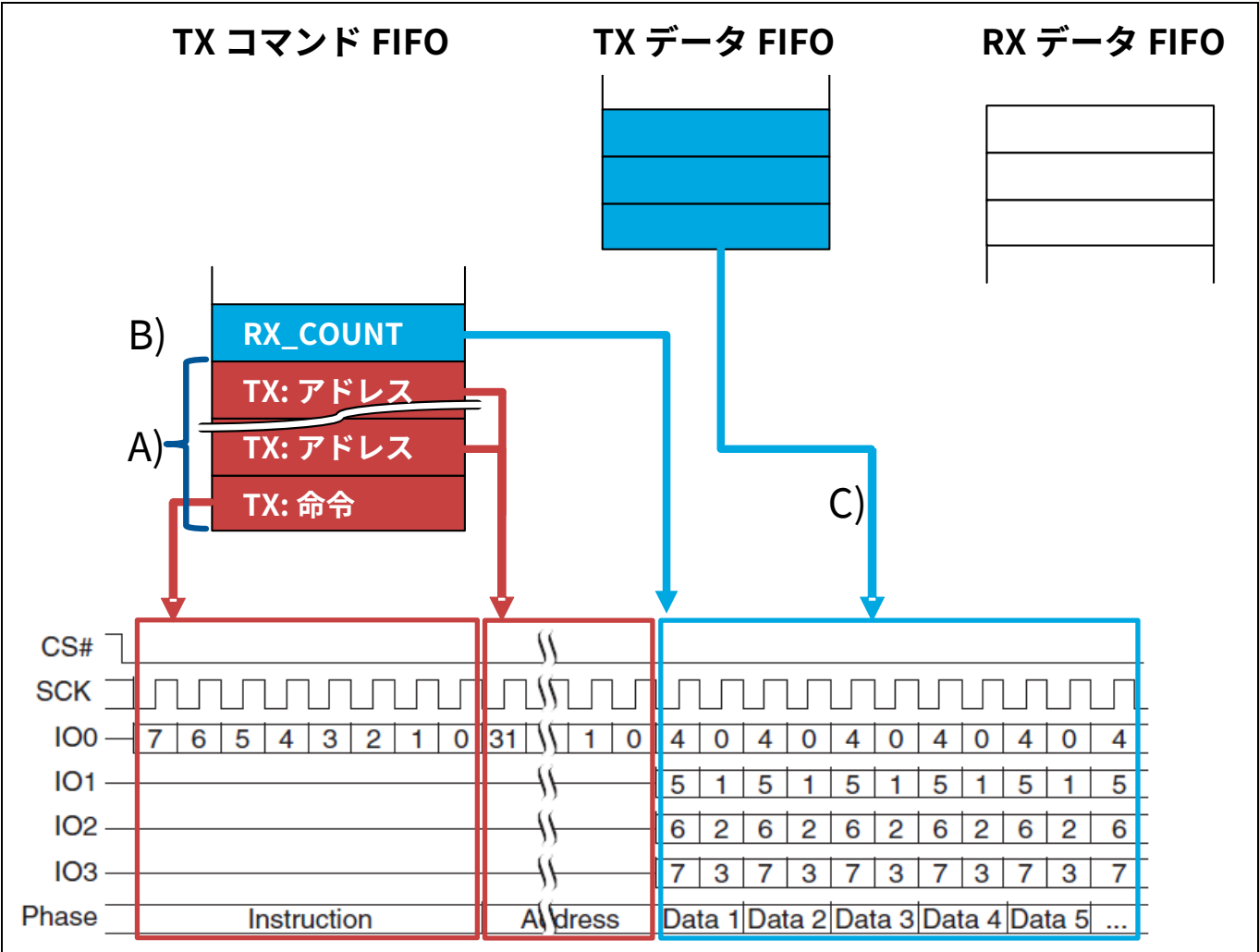


Figure 3 クアッド SPI のプログラムコマンドの構築

クアッド SPI のプログラムコマンドの TX コマンド FIFO の設定を以下に説明します。

- A) “TX” コマンドを使用して、命令とアドレスの順序で TX コマンド FIFO にエントリを設定。
- B) “TX_COUNT” コマンドを使用して、送信データの数 TX コマンド FIFO に設定。
- C) SMIF は TX データ FIFO からデータセットの数を送信。

Table 6 に、クアッドページプログラム (4QPP 34h) の TX コマンド FIFO エントリの詳細を示します。

概要

2.4 データキャプチャ方式

SMIF は次のデータキャプチャをサポート

- 出力/フィードバッククロックベースキャプチャ
- 内部クロックベースキャプチャ
- RWDS ベースキャプチャ
- 遅延ラインおよびデータ学習パターンベースキャプチャ

以下は、各キャプチャ方式の概要です。詳細については [architecture TRM](#) を参照してください。

2.4.1 出力/フィードバッククロックベースキャプチャ

このキャプチャ方式は、SMIF 出力または SDR および DDR タイミングの出力フィードバッククロックでデータをキャプチャします。キャプチャクロックとして、メモリ出力クロック (spihb_clk_in), 反転メモリ出力クロック, メモリ出力フィードバッククロック (spihb_clk_in), または反転メモリ出力フィードバッククロックを使用します。この方式には、出力またはフィードバックを遅延させてサンプル時間をより細かく調整するための遅延ラインがあります。クロックは、SMIFx_CTL レジスタの CLOCK_IF_RX_SEL[3:0] で選択できます。

出力クロックは、CYT4B および CYT3B シリーズのみで選択できます。

2.4.2 内部クロックベースキャプチャ

このキャプチャ方式では、インタフェースクロック (clk_if) または反転インタフェースクロックをキャプチャクロックとして使用します。この方式は、出力/フィードバッククロックベースのキャプチャ方式と比較して、常に内部クロックのキャプチャクロックのソースとして利用できます。クロックは、SMIFx_CTL レジスタの CLOCK_IF_RX_SEL[3:0] で選択できます。

2.4.3 RWDS ベースキャプチャ

このキャプチャ方式では、RWDS 信号をクロックとして使用して、入力データをキャプチャします。出力またはフィードバックを遅延させてサンプル時間をより細かく調整するための遅延ラインがあります。クロックは、SMIFx_CTL レジスタの CLOCK_IF_RX_SEL[3:0] で選択できます。

概要

2.4.4 遅延ラインおよびデータ学習パターン (DLP) ベースキャプチャ

このキャプチャ方式では、内部クロック (clk_if または反転 clk_if) をクロックとして使用して、入力データをキャプチャします。このキャプチャ方式は、キャプチャラインを正確に調整するための遅延ラインを提供します。

遅延ラインタップの選択は、ソフトウェアまたはデータ学習を介したハードウェアで自動的行えます。データ学習方式は、キャプチャされたデータ学習パターンを期待されるパターンと比較することで、各データ入力ラインに対してハードウェアで最適な遅延ラインタップを見つけます。入力データは、データ学習パターンの後に受信します。

この方式では、ソフトウェアは MMIO モードでメモリ読出しトランザクションを開始する必要があります。

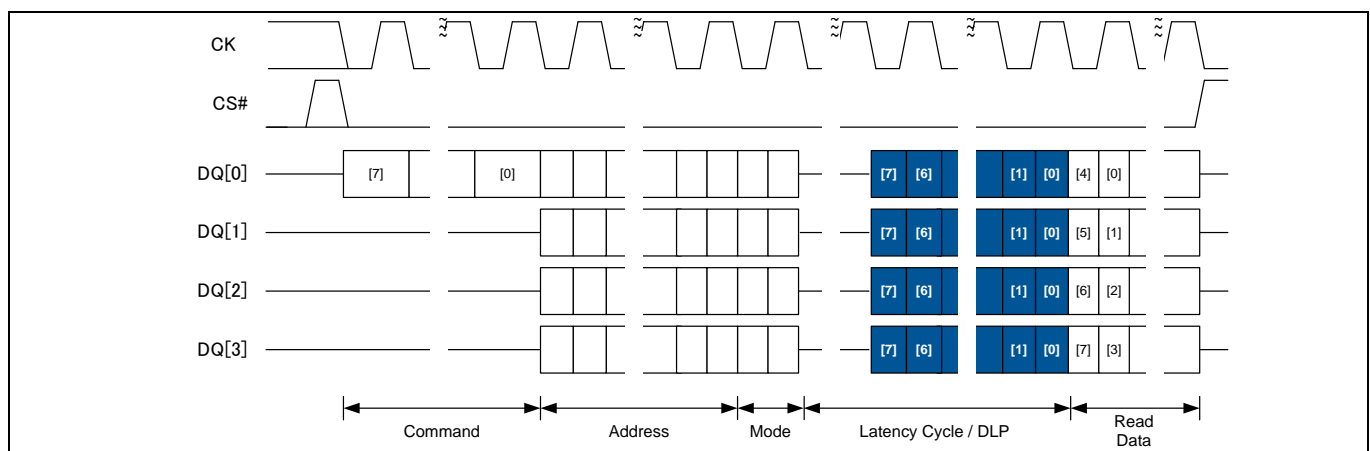


Figure 4 DLP 出力イメージ

メモリデバイスは、要求された読出しデータが提供される前の読出しレイテンシサイクル内で、すべてのデータ I/O ピンに既知のデータパターン (データ学習パターン) を提供します。

この方式は、CYT4B および CYT3B シリーズには対応しません。

デュアルクアッド SPI DDR モードの例

3 デュアルクアッド SPI DDR モードの例

このセクションでは、データの書込み (SDR) およびデータの読出し (DDR) について説明します。MMIO モードでの暗号化は、データの書込みに使用し、XIP モードは、データの読出しに使用します。デュアルクアッド SPI モードでは、接続方法についても説明します。

3.1 メモリデバイスの接続

この図では、SPI 接続の例としてデュアルクアッド SPI モードを使用します。Figure 5 では、メモリデバイス 0 と 1 を使用し、どちらもクアッド SPI メモリです。各デバイスは専用のデータ信号を接続します。TRAVEO™ T2G アドレス空間でのデバイスアドレス領域は同じで、SPIHB_SEL0 と SPIHB_SEL1 の出力は同じです。これは、デュアルクアッド設定として知られます。SPI 読出しおよび書込み転送中、各デバイスは 1 バイトのニブルを提供します。

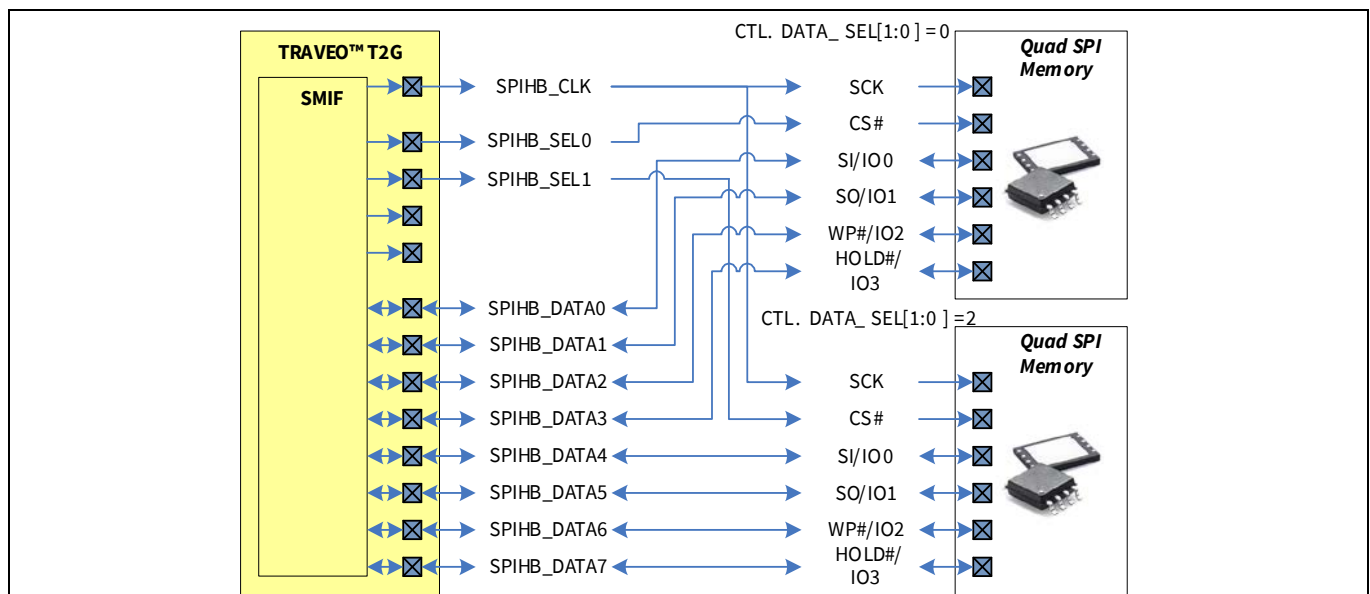


Figure 5 デュアルクアッド設定

3.2 デュアルクアッドモードでの S25FL256 デバイスの使用

この例では、クアッド SPI フラッシュメモリ S25FL256S を使用して、消去、書込み (SDR), および読出し (DDR) がチェックされます。Figure 6 に、メモリのコマンドの流れを示します。このフローは、暗号化を使用した MMIO モードのデュアルクアッド SPI モードを使用して、16 バイトのデータをデバイスに書込みます。書込まれている 16 バイトのデータはランダムデータです。読出しの場合、このフローは、MMIO モードでのクアッド I/O 読出し (4QIOR ECh) および XIP モードでのオンザフライでの復号化による読出し転送のモードをデバイスに設定します。Figure 6 のクアッドのすべてのコマンドは、デュアルクアッドモードです。

デュアルクアッド SPI DDR モードの例

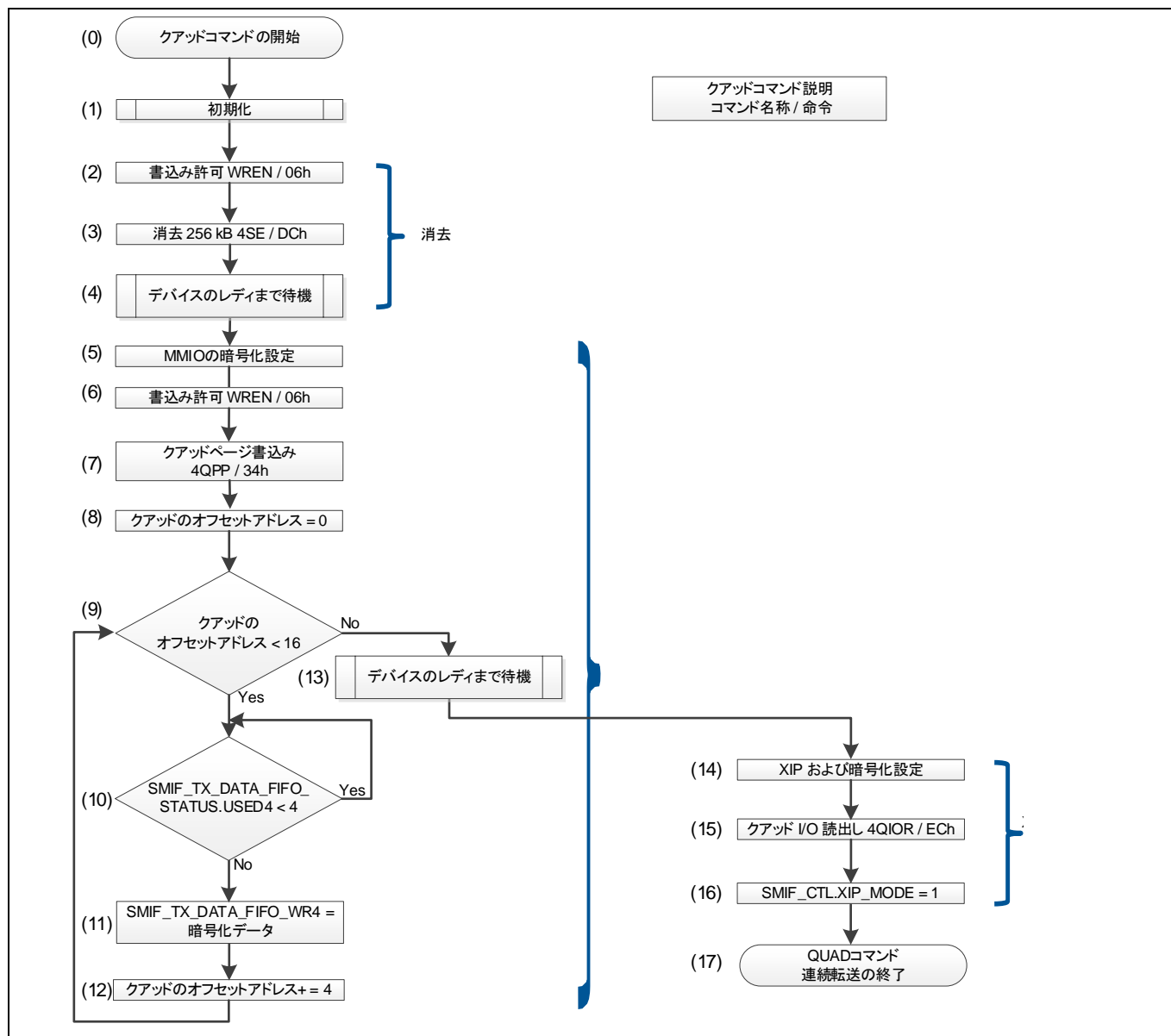


Figure 6 クアッドのコマンドを使用したソフトウェアフローチャートの例

デュアルクアッド SPI DDR モードの例

ソフトウェアのフローチャートの例を以下に説明します。

- (0) この例では、デュアルクアッド SPI モードを設定。
- (1) SMIF およびデュアルクアッドのデバイスの初期化。 [3.3 初期化](#) 参照。
ここから (4) までは、クアッドの 4SE コマンドを使用した消去操作の一部です。
- (2) 4SE コマンドの WREN コマンドを送信。 [3.4 消去](#) 参照。
- (3) 4SE コマンドを送信して、アドレス指定されたセクタのすべてのビットを 1 に設定。 [Table 5](#) 参照。
- (4) デバイスがレディまで待機。 [Figure 9](#) 参照。
ここから (12) までは、クアッドの 4QPP コマンドを使用したプログラム操作の一部です。
- (5) MMIO の暗号化設定を設定。 [Figure 13](#) 参照。
- (6) 4QPP コマンドの WREN コマンドを送信。
- (7) 書込むため 4QPP コマンドを送信。 [Table 6](#) 参照。
例で使用される各クアッドデバイスのセクタのアドレス: 0x01240000
サイズ: 16 バイト
- (8) FOR LOOP のオフセットアドレスを初期化。
- (9) オフセットアドレスで暗号化されたデータの量をチェック。
クアッドのオフセットアドレス < 16 の場合、(10) に進む。
それ以外の場合は、(13) に進む。
- (10) SMIF_TX_DATA_FIFO_STATUS.USED4 ビットが 5 以上になるまで待機。
- (11) SMIF_TX_DATA_FIFO_WR4 を暗号化されたデータに設定。
- (12) オフセットアドレスに 4 を追加し、(9) に進む。
- (13) デバイスがレディまで待機。
- (14) XIP の設定と暗号化設定を設定。 [3.6 XIP モードでの読出し転送](#) 参照。
- (15) 4QIOR コマンドを送信して、クアッド I/O ハイパフォーマンス読出しモードを設定。
- (16) SMIF_CTL.XIP_MODE を “1” に設定 (XIP モード)。
- (17) メモリはクアッド I/O ハイパフォーマンス読出しモードのままであり、命令を必要とせずに次のアドレスを入力できます (CS# を HIGH に上げてから LOW にアサートした後)。この後、ハードウェアは、AHB-Lite または AXI 読出し転送のメモリ読出し転送を自動的に生成します。

デュアルクアッド SPI DDR モードの例

3.3 初期化

この場合、SMIF とデバイス、[Figure 6](#) の (1) に示すように初期化されます。[Figure 7](#) は、初期化のフローを示します。

フローチャートでは、SMIF が初期化され、デュアルクアッドのデバイスにリセットが送信されます。

デバイスのコンフィギュレーションレジスタは、[Figure 7](#) の (5) および (6) のように RDCR (コンフィギュレーションレジスタの読出し) コマンドによって確認されます。4 ビット幅のクアッド I/O がデバイスに設定されていない場合、クアッドビットは "1" に設定されます。

デバイス指定のないクアッドのコマンド (例えば (2)) は、デュアルクアッドモードです。
“SMIF_TX_CMD_FIFO_WR”の“Device select”ビットに複数のビット、例えば 0x3 ([Table 2](#), [Table 5](#)) が設定される場合、コマンドはデュアルクアッドモードです。デバイス仕様 (例えば (5)) を持つクアッドのコマンドはクアッドモードです。“Device select”に 1 ビットのみが設定される場合、例えば 0x1 または 0x2 ([Table 3](#), [Table 4](#)) コマンドはクアッドモードです。

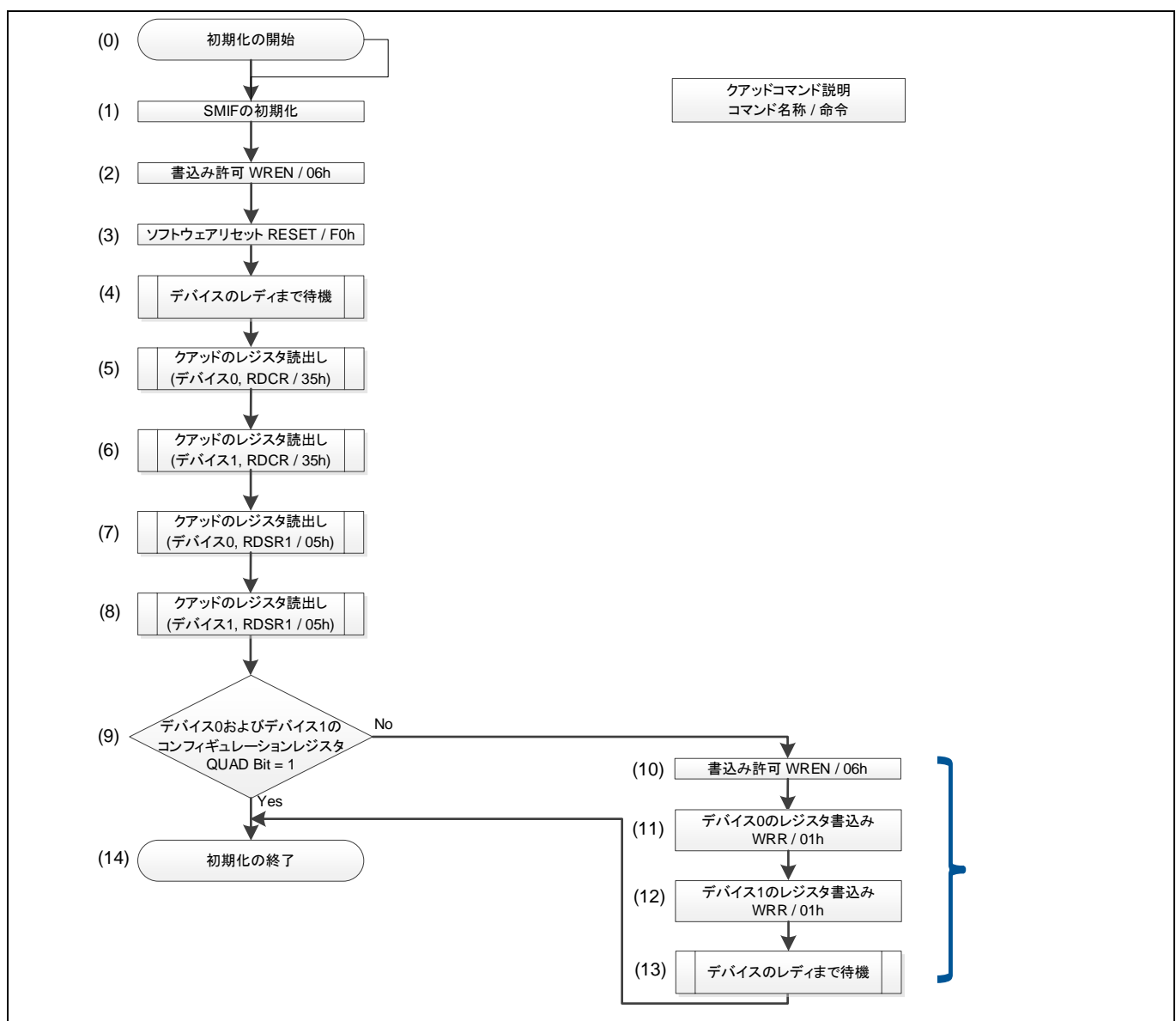


Figure 7 初期化のソフトウェアフローチャート

デュアルクアッド SPI DDR モードの例

初期化のソフトウェアフローを以下に説明します。

- (0) SMIF およびデュアルクアッドのデバイスの初期化を設定。
- (1) SMIF を初期化して、クアッドのコマンドを送信。3.3.1 SMIF の初期化参照。
- (2) WREN コマンドを送信して、WRR コマンドを後で実行できるようにする。Table 2 参照。
- (3) RESET コマンドを送信してメモリをリセットし新しいコマンドを待機。Table 2 参照。
- (4) デバイスの準備が整うまで待ちます。Figure 9 参照。
- (5) RDCR (read configuration register) コマンドを送信してデバイス 0 のクアッドビットをチェック。Table 3 参照。
- (6) RDCR (read configuration register) コマンドを送信して、デバイス 1 のクアッドビットをチェック。
- (7) RDSR1 (read status register-1) コマンドを送信して、デバイス 0 の BP ビットをチェック。Table 3 参照。
- (8) RDSR1 (read status register-1) コマンドを送信して、デバイス 1 の BP ビットをチェック。
- (9) クアッドビットが 1 に設定されている場合、このビットはデバイスのデータ幅を 4 ビット - クアッドモードに切り替え。
両方のクアッドビットが“1”ではない場合、(10)に進む。
両方のクアッドビットが“1”の場合、(14)に進む。
- (10) WREN コマンドを送信して、WRR コマンドを後で実行できるようにする。Table 2 参照。
- (11) WRR コマンドを送信して、デバイス 0 のクアッドビットと LC ビットを設定。Table 4 参照。
- (12) WRR コマンドを送信して、デバイス 1 のクアッドビットと LC ビットを設定。
- (13) デバイスがレディまで待機。
- (14) SMIF およびデュアルクアッドのデバイスの初期化が完了。

3.3.1 SMIF の初期化

SMIF は Figure 7 の (1) で示されるように初期化に設定されます。

デバイス 0 および 1 は、デュアルクアッド SPI モードを実装するために使用されます。registers TRM の SMIF0_DEVICE の章では、デバイス 0 と 1 は SMIF0_DEVICE0 と SMIF0_DEVICE1 として定義されます。SMIF0_DEVICE0 および SMIF0_DEVICE1 にはそれぞれ SMIF_DEVICE_CTL レジスタがあります。このアプリケーションノートでは、レジスタは SMIF_DEVICE0_CTL および SMIF_DEVICE1_CTL として示されます。SMIF0_DEVICE0 と SMIF0_DEVICE1 の両方が設定されている場合、レジスタは SMIF_DEVICE_x_CTL のように表示されます。設定手順は以下のとおりです。

1. クロッキングシステムの設定^[1]
2. ポート設定
3. SMIF ブロックを通信ブロックとして初期化
 - SMIF_INTR_MASK.TR_TX_REQ を“0”に設定 (無効: 初期割込みマスクを設定)
 - SMIF_INTR_MASK.TR_RX_REQ を“0”に設定 (無効: 初期割込みマスクを設定)
 - SMIF_CTL.CLOCK_IF_TX_SEL を“1”に設定 (DDR)
 - SMIF_CTL.CLOCK_IF_RX_SEL を“1”に設定 (DDR または SDR キャプチャ用の SMIF 出力反転クロック)
 - SMIF_CTL.DELAY_TAP_ENABLED を“0”に設定 (レジスタ DELAY_TAP_SEL または INT_CLOCK_DELAY_TAP_SEL0/1 は未使用)
 - SMIF_CTL.INT_CLOCK_DL_ENABLED を“1”に設定 (有効。遅延ラインタップの選択は、データ学習パターンに基づいた HW によって変更)
 - SMIF_CTL.XIP_MODE を“0”に設定 (MMIO モード)

¹ 設定の詳細については、architecture TRM の関連する章を参照してください。

デュアルクアッド SPI DDR モードの例

4. SMIF デバイス 0/1 の設定

- Set SMIF_DEVICE0_CTL.DATA_SEL を “0” に設定 (spi_data[0] = IO0, spi_data[1] = IO1, ..., spi_data[7] = IO7。
- この値はシングル, デュアル, クアッド, デュアルクアッドおよびオクタル SPI モードで使用可能。この値はデュアルクアッド SPI モードの最初のデバイスに使用する必要あり)
- SMIF_DEVICE0_CTL.WR_EN を “1” に設定 (このデバイスへの書き込み転送許可)
- SMIF_DEVICE1_CTL.DATA_SEL を “2” に設定 (spi_data[4] = IO0, spi_data[5] = IO1, ..., spi_data[7] = IO3。この値はシングル, デュアル, クアッドおよびデュアルクアッド SPI モードのみで許可。デュアルクアッド SPI モードでは、この値を 2 番目のデバイスに使用する必要あり)
- SMIF_DEVICE1_CTL.WR_EN を “1” に設定 (このデバイスへの書き込み転送許可)

5. SMIF の有効化

- SMIF_CTL.ENABLED を “1” に設定 (有効)
- SMIF_CTL の読出し (レジスタを読出しバッファをフラッシュ)

3.3.2 クアッドのリセット

RESET コマンドは **Figure 7** の (3) での、デュアルクアッドのデバイスをリセットするために送信されます。RESET コマンドを送信する前に、クアッドの WREN コマンドを送信する必要があります。WREN コマンド送信後、デバイスはステータスレジスタに書き込みイネーブルラッチ (WEL) を設定して、書き込み操作を有効にします。RESET および WREN は **Figure 8** に示すように、命令のみで構成されるスタンドアロンの命令コマンドです。WRR コマンドの詳細については、**S25FL256S のデータシート** を参照してください。

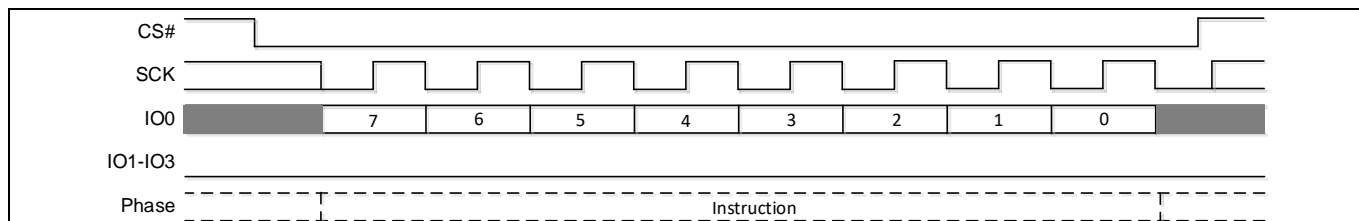


Figure 8 スタンドアローン命令コマンド

Table 2 に、スタンドアローン命令コマンドの TX コマンド FIFO エントリを示します。デュアルクアッド SPI モードのため、spi_select[0] および spi_select[1] の両方が選択されます。

Table 2 スタンドアローン命令コマンドの SMIF_TX_CMD_FIFO_WR 設定

ビット	26:24	23	22	21	20	19	18	17:16	15:18 ^[2]	7:0	SMIF_TX_CMD_FIFO_WR
エントリ	コマンド	デバイス選択				最終 TX	SDR/DDR	データ転送幅		送信バイト	
1	0 (命令)	3				1	0	0	0	8 ビット 命令	0x038_00xx

RESET コマンドの場合、8 ビット命令を “0xF0” に設定します。WREN コマンドの場合、**Table 2** で指定されている 8 ビット命令を “0x01” に設定します。

² "TX" コマンドの場合、DATA[15:8] は 2 番目に送信されるバイトを指定します。これは、DDR モードでのオクタルデータ転送、つまり DATA[17:16] = "3" および DATA[18] = "1" の場合にのみ使用されます (指定の必要あり)。

デュアルクアッド SPI DDR モードの例

Figure 7 のステップ **(4)** では、デバイスの WIP ビットをチェックして RESET コマンドの完了を確認します。

Figure 9 は“デバイスがレディまで待機”のフローを示します。

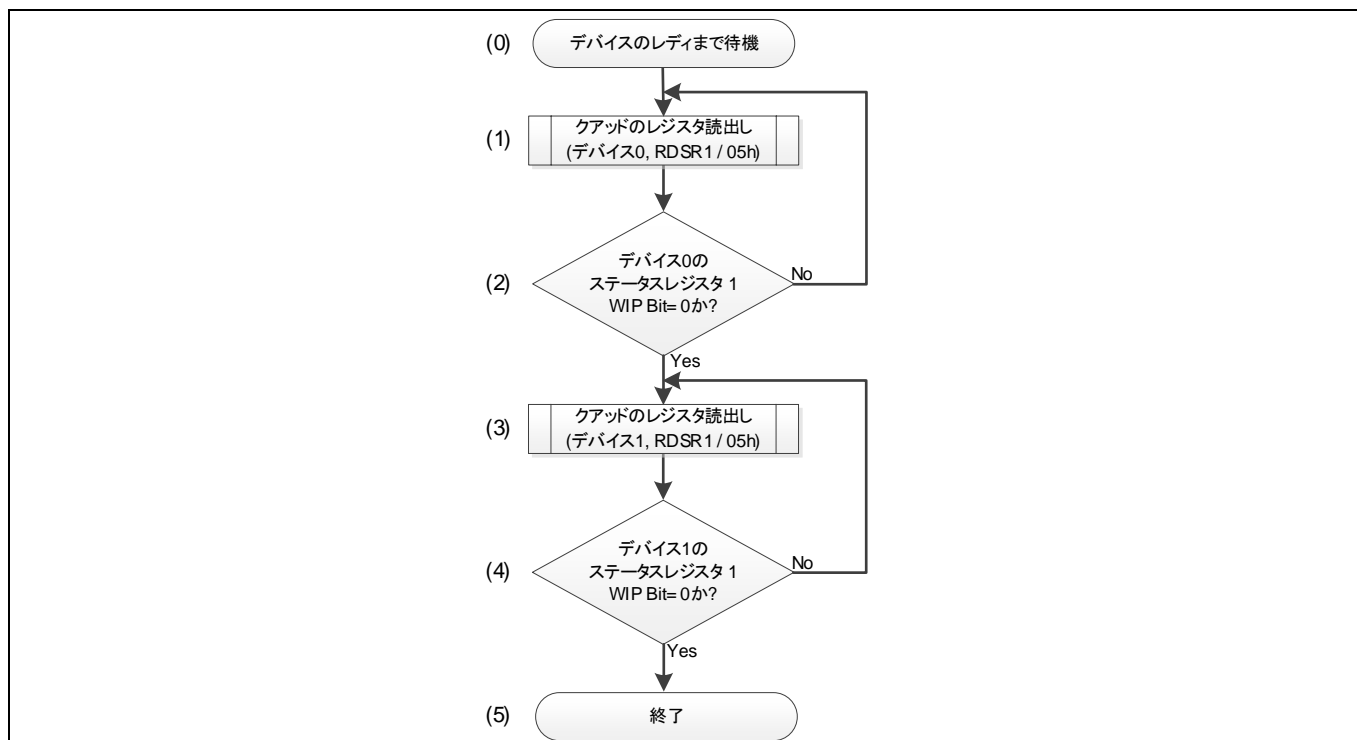


Figure 9 デバイスがレディまで待機のソフトウェアフローチャート

デバイスがレディまで待機するソフトウェアフローを以下に説明します。

- (0) “デバイスがレディまで待機”のフローチャート開始。
- (1) RDSR1 (Read Status Register-1) コマンドを送信して、デバイス 0 の WIP ビットをチェック。Table 3 を参照。
- (2) WIP ビットが 1 に設定されている場合、デバイスはビジー。
WIP ビットが “0” ではない場合、(1) に進む。
WIP ビットが “0” の場合、(3) に進む。
- (3) RDSR1 (Read Status Register-1) コマンドを送信して、デバイス 1 の WIP ビットをチェック。
- (4) WIP ビットが 1 に設定されている場合、デバイスはビジー。
WIP ビットが “0” ではない場合、(3) に進む。
WIP ビットが “0” の場合、(5) に進む。
- (5) “デバイスがレディまで待機”のフローチャート終了。

3.3.3 クアッドのレジスタ読出し

関数 "クアッドのレジスタ読出し" は [Figure 7](#) と [Figure 9](#) の両方で使用されます。この関数は、引数で設定されたクアッドのレジスタ値を取得します。[Figure 10](#) は、“クアッドのレジスタ読出し”のフローチャートです。

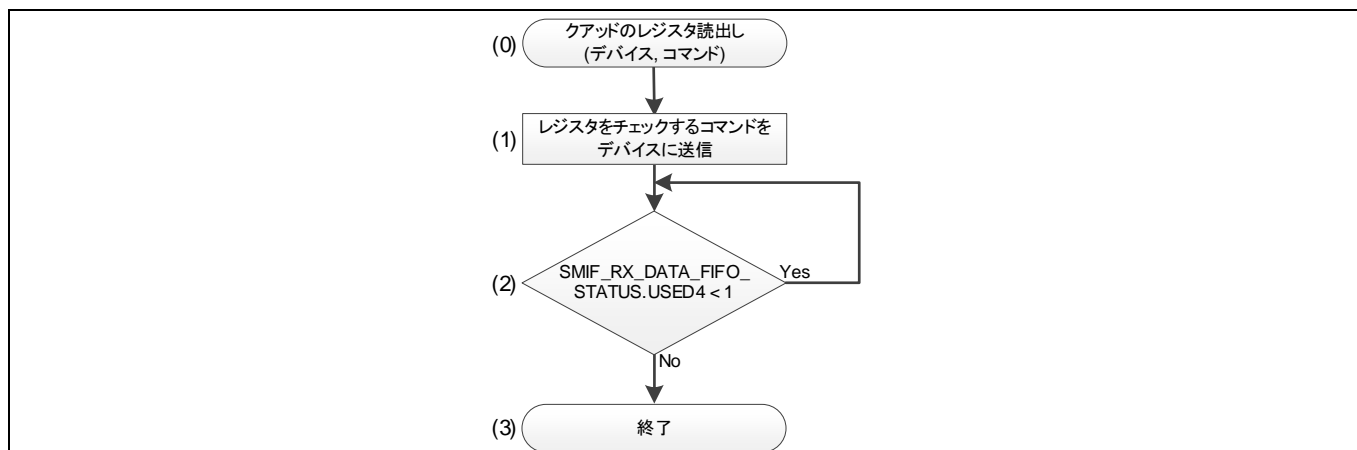


Figure 10 クアッドのレジスタ読出しのソフトウェアフローチャート

- (0) クアッドのレジスタの読出し。2つの引数には、デバイスとコマンドを含む。
- (1) コマンド引数を設定されたデバイスに送信。
- (2) SMIF_RX_DATA_MMIO_FIFO_STATUS.USED4 ビットが1以上になるまで待機。
- (3) 終了。FIFO から1バイトのみを読出す必要がある場合、SMIF_RX_DATA_MMIO_FIFO_RD1 レジスタをチェック。

RDCR および RDSR1 コマンドは、[Figure 11](#) に示すようにレジスタ読出しコマンドシーケンスです。

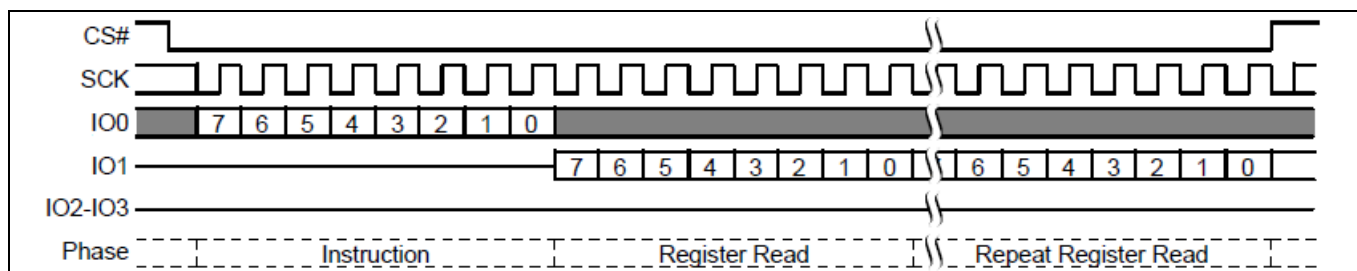


Figure 11 レジスタの読出しコマンドシーケンス

デュアルクアッド SPI DDR モードの例

Table 3 に、レジスタ読出しコマンドシーケンスの TX コマンド FIFO エントリを説明します。このアプリケーションノートでは、クアッドフラッシュのレジスタを讀出すために、spi_select[0] および spi_select[1] の SMIF_TX_CMD_FIFO_WR 設定が個別に設定されます。

Table 3 レジスタ読出しコマンドシーケンスの SMIF_TX_CMD_FIFO_WR 設定

ビット	26:24	23	22	21	20	19	18	17:16	15:18 ^[3]	7:0	SMIF_TX_CMD _FIFO_WR
エントリ	コマンド	デバイス選択				最終 TX	SDR/ DDR	データ 転送幅		送信バイト	
デバイス 0											
1	0 (命令)	1 (spi_select_out[0])				0	0	0	0	8 ビット 命令	0x010_00xx
2	2 (RX_COUNT)	1				1	0	0	0x0000 (受信メモリ データユニット ^[4])		0x218_0000
デバイス 1											
1	0 (命令)	2 (spi_select_out[1])				0	0	0	0	8 ビット 命令	0x020_00xx
2	2 (RX_COUNT)	2				1	0	0	0x0000 (受信メモリ データユニット ^[4])		0x228_0000

RDCR コマンドの場合、8 ビット命令を “0x35” に設定。RDSR1 コマンドの場合、8 ビット命令を “0x05” に設定。

³ "TX" コマンドの場合、DATA[15:8] は 2 番目に送信されるバイトを指定します。これは、DDR モードでのオクタルデータ転送、つまり DATA[17:16] = "3" および DATA[18] = "1" の場合にのみ使用されます (指定の必要あり)。

⁴ "RX_COUNT" コマンドの場合、DATA [15:0] は受信したメモリデータユニットの数 (マイナス 1) を指定します。“0”:1 ユニット、“1”:2 ユニット。SPI (DDR を使用する 8 進 SPI を除く) の場合、1 つのメモリデータユニットは 1 バイトであり、DDR を使用する 8 進 SPI の場合、1 つのメモリデータユニットは 2 バイトワードです。使用される RX データ FIFO エントリの数 (RX_DATA_FIFO_STATUS 内) は、受信するメモリデータユニットの数*8/データ幅 (1, 2, 4, 8) と同じです。使用される RX データ MMIO FIFO エントリの数 (RX_DATA_MMIO_FIFO_STATUS 内) は、バイト数と同じです。

デュアルクアッド SPI DDR モードの例

3.3.4 クアッドのレジスタ設定

Figure 7 のステップ (10) から (13) は、クアッドのレジスタを設定します。Figure 7 の (11) および (12) に示すように、WRR コマンドを送信してデバイスのレジスタを設定します。WRR コマンドを送信する前に、RESET コマンドのようなクアッドの WREN コマンドを送信する必要があります。

Table 4 に、WRR コマンドの TX コマンド FIFO エントリを説明します。このアプリケーションノートでは、spi_select[0] および spi_select[1] の SMIF_TX_CMD_FIFO_WR 設定は WRR コマンドに対して個別に設定されます。Figure 9 は、デバイスの WIP ビットをチェックして、WRR コマンドの完了を確認します。

Table 4 書き込みレジスタの SMIF_TX_CMD_FIFO_WR 設定 (WRR 01h)

ビット	26:24	23	22	21	20	19	18	17:16	15:18 ^[5]	7:0	SMIF_TX_CMD _FIFO_WR
エントリ	コマンド	デバイス選択			最終 TX	SDR/ DDR	データ 転送幅	送信バイト			
デバイス 0											
1	0 (命令)	1 (spi_select_out[0])			0	0	0	0	0x01	0x010_0001	
2	0 (入力データ)	1			0	0	0	ステータスレジスタ 1 ^[6] で BP ビット = 0 設定		0x010_00xx	
3	0 (入力データ)	1			1	0	0	コンフィグレーションレジスタ ^[6] で QUAD ビット = 1 LC ビット = 1 設定		0x018_00xx	
デバイス 1											
1	0 (命令)	2 (spi_select_out[1])			0	0	0	0	0x01	0x020_0001	
2	0 (入力データ)	2			0	0	0	ステータスレジスタ 1 ^[6] で BP ビット = 0 設定		0x020_00xx	
3	0 (入力データ)	2			1	0	0	コンフィグレーションレジスタ ^[6] で QUAD ビット = 1 LC ビット = 1 設定		0x028_00xx	

⁵ "TX" コマンドの場合、DATA[15:8] は 2 番目に送信されるバイトを指定します。これは、DDR モードでのオクタルデータ転送、つまり DATA[17:16] = "3" および DATA[18] = "1" の場合にのみ使用されます (指定の必要あり)。

⁶ Table 3 に示すように、事前に各クアッドのレジスタ値を取得してください。次に、レジスタ値に必要なビットを設定してください。

デュアルクアッド SPI DDR モードの例

3.4 消去

Figure 6 のステップ (2) から (4) は、クアッドの消去操作を示します。Figure 6 の (3) に示すように、4SE コマンドを送信してデバイスのセクタを消去します。4SE コマンドを送信する前に、クアッドの WREN コマンドを送信する必要があります。

Table 5 に、4SE コマンドの TX コマンド FIFO エントリを説明します。

Table 5 セクタ消去の SMIF_TX_CMD_FIFO_WR 設定 (4SE DCh)

ビット	26:24	23	22	21	20	19	18	17:16	15:18 ^[7]	7:0	SMIF_TX_CMD_FIFO_WR
エントリ	コマンド	デバイス選択				最終 TX	SDR/DDR	データ転送幅		送信バイト	
1	0 (命令)	3				0	0	0	0	0xDC	0x030_00DC
2	0 (アドレス)	3				0	0	0	0	0x01	0x030_0001
3	0 (アドレス)	3				0	0	0	0	0x24	0x030_0024
4	0 (アドレス)	3				0	0	0	0	0x00	0x030_0000
5	0 (アドレス)	3				1	0	0	0	0x00	0x030_0000

Figure 8 は、デバイスの WIP ビットをチェックして、4SE コマンドの完了を確認します。

3.5 MMIO モードでの書込み転送

Figure 6 のステップ (5) から (13) は、クアッドをプログラムします。MMIO モードはステップ (5) から (13) で使用されます。クアッドのコマンドを送信する前に、暗号化を設定する必要があります。

XIP モードでは、暗号化コンポーネントは書込みデータのオンザフライ暗号化と読み出しデータのオンザフライ暗号化に対応します。オンザフライ暗号化の使用は、デバイスの SMIF_DEVICEx_CTL.CRYPTO_EN によって決定されます。Figure 12 に、完全な XIP モード機能を示します。

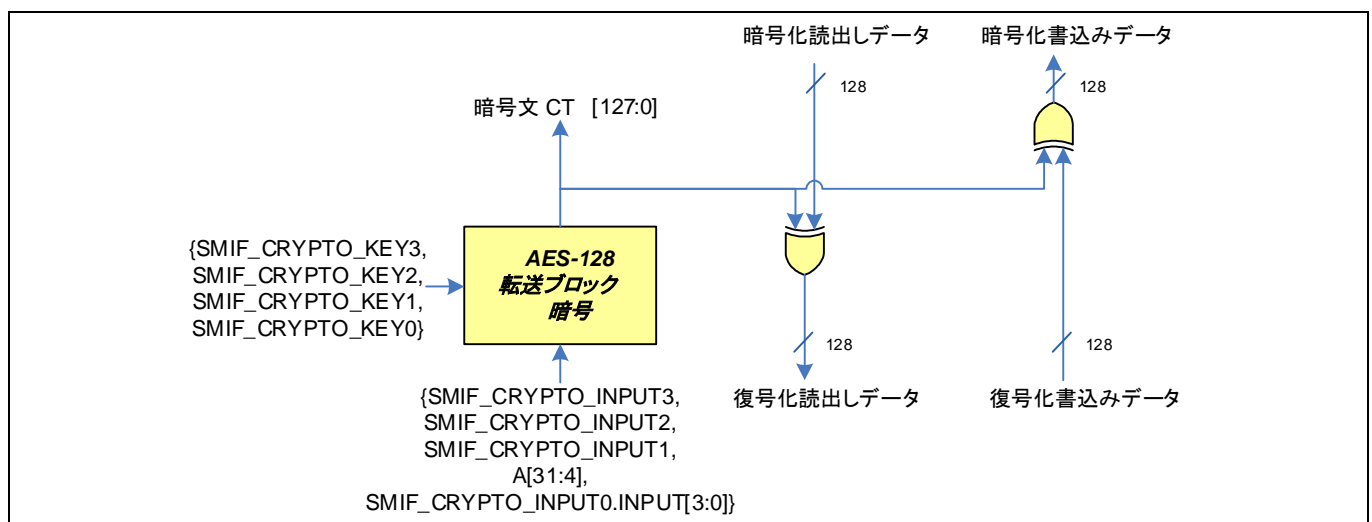


Figure 12 XIP モードでの暗号化

⁷ “TX”コマンドの場合、DATA[15:8]は2番目に送信されるバイトを指定します。これは、DDR モードでのオクタルデータ転送、つまり、DATA[17:16] = “3” および DATA[18] = “1” の場合にのみ使用されます(指定する必要あり)。

デュアルクアッド SPI DDR モードの例

MMIO モードでは、MMIO レジスタインタフェースを介して暗号化コンポーネントにアクセスし、オフラインの暗号化と復号化に対応します。この例では、メモリで 4QPP コマンドを使用してプログラミングする前に、**Figure 6** の (5) に示すように、MMIO モードでプログラムデータの暗号化が設定されます。

Figure 13 に、MMIO モードでのプログラムデータの暗号化を示します。XIP モードの暗号化では、フローはハードウェアによって実現されます。

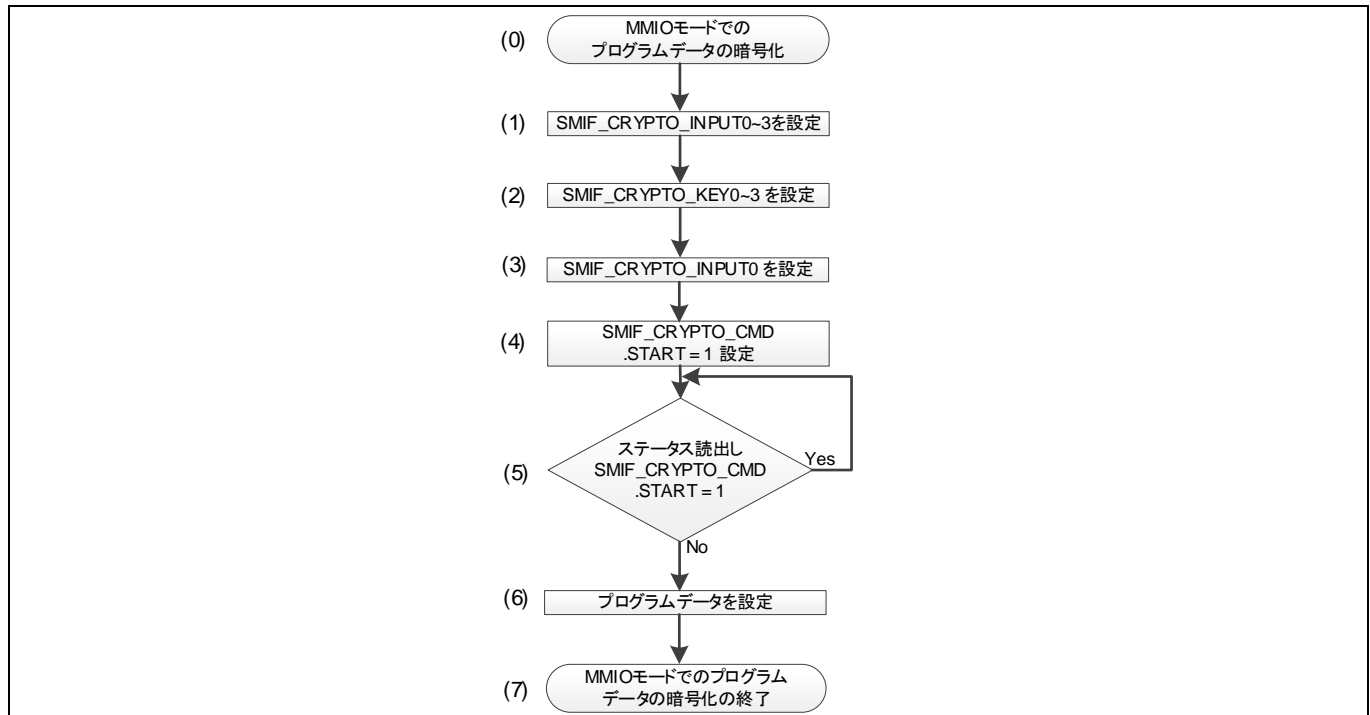


Figure 13 MMIO モードでのプログラムデータの暗号化ソフトウェアのフローチャートの例

ソフトウェアフローの例を以下に説明します。

(0) この例では、MMIO モードでの暗号化を設定。

(1) SMIF_CRYPTO_INPUT0~3 をプレーンテキスト PT[127:0] に設定 (プレーンテキスト PT[127:0] は AES-128 転送ブロック暗号への入力)。SMIF_CRYPTO_INPUT0[0:31], SMIF_CRYPTO_INPUT1[0:31], SMIF_CRYPTO_INPUT2[0:31], SMIF_CRYPTO_INPUT3[0:31] は静的な値です。これは“ノンス”に対応することを意味します。SMIF_CRYPTO_INPUT0[4:31] は、読出し/書込みアドレスに従って動的に設定されます。これはカウンタに対応します。したがって、これはブロック暗号の CTR モードに似ています。暗号化コンテキストのノンスは、通常、暗号化操作に 1 度だけ使用される任意の番号です。

(2) SMIF_CRYPTO_KEY0~3 を秘密鍵 KEY[127:0] に設定 (鍵 KEY[127:0] は AES-128 転送ブロック暗号鍵)。

(3) SMIF_CRYPTO_INPUT0 を以下の値に設定。

Value = ((0x6000_0000 + 0x0124_0000 x 2) & 0xFFFF_FFF0) + (SMIF_CRYPTO_INPUT0 & 0x0000_000F)

Cortex®-M7 および Cortex®-M0+ のアドレスマップの SMIF 開始アドレス範囲: 0x6000_0000

例で使用される各クアッドデバイスのセクタアドレス: 0x0124_0000

(4) SMIF_CRYPTO_CMD.START を “1” に設定 (SW はこのフィールドを '1' に設定して AES-128 転送ブロック暗号操作を開始します。ハードウェアはこのフィールドを '0' に設定して操作の完了を示します。)。

(5) SMIF_CRYPTO_CMD.START ビットが “0” に設定されるまでチェック。

(6) プログラムデータを以下に設定。

Program data = (Input data) XOR (SMIF_CRYPTO_OUTPUT0~3)

(7) この例では、MMIO モードでの暗号化の終了を設定。

デュアルクアッド SPI DDR モードの例

Figure 6 の (7) のように、書込むために 4QPP コマンドがデバイスへ送信されます。送信する前に、クアッドの WREN コマンドを送信する必要があります。**Table 6** は、クアッドページプログラム (4QPP 34h) の TX コマンド FIFO エントリを説明します。**Figure 9** はデバイスの WIP ビットをチェックして、4QPP コマンドの完了を確認します。

Table 6 クアッドページプログラムの SMIF_TX_CMD_FIFO_WR 設定 (4QPP 34h)

ビット	26:24	23	22	21	20	19	18	17:16	15:18 ^[8]	7:0	SMIF_TX_CMD_FIFO_WR
エントリ	コマンド	デバイス選択				最終 TX	SDR/DDR	データ転送幅		送信バイト	
1	0 (命令)	3				0	0	0	0	0x34	0x030_0034
2	0 (アドレス)	3				0	0	0	0	0x01	0x030_0001
3	0 (アドレス)	3				0	0	0	0	0x24	0x030_0024
4	0 (アドレス)	3				0	0	0	0	0x00	0x030_0000
5	0 (アドレス)	3				0	0	0	0	0x00	0x030_0000
6	1 (TX_COUNT)	3				1	0	3	0x000F (送信メモリ データユニット ^[9])		0x030_0000

3.6 XIP モードでの読出し転送

XIP モードの SMIF は (ソフトウェアの介入無しで) TX FIFO および RX FIFO へのアクセスにより、メモリ転送を自動的に生成します。AHB-Lite または AXI 読出し/書込み転送のメモリ読出し/書込み転送を生成します。

これは、次の XIP ブロックで行われます。

- AHB-Lite または AXI インタフェースからの読出しまたは書込み転送要求を TX コマンド FIFO のコマンドに変換
- TX/RX データ FIFO との間でデータの送受信

この例では、クアッドフラッシュの機能であるクアッド I/O ハイパフォーマンス読出しモードが使用されます。**Figure 6** の (14) で示されるデュアルクアッド設定の場合、spi_data[3:0] に接続されたデバイス 0 と spi_data[7:4] に接続されたデバイス 1 で設定が異なり、デバイス 0 とデバイス 1 の両方が各レジスタを設定する必要があります。XIP モードによる設定を以下に説明します。

6. SMIF_DEVICE0_CTL.DATA_SEL を “0” に設定 (spi_data[0] = IO0, spi_data[1] = IO1, ..., spi_data[7] = IO7)
SMIF_DEVICE1_CTL.DATA_SEL を “2” に設定 (spi_data[4] = IO0, spi_data[5] = IO1, ..., spi_data[7] = IO3)
7. SMIF_DEVICEx_CTL.WR_EN を “1” に設定 (このデバイスへの書込み転送許可)
8. SMIF_DEVICEx_CTL.CRYPTO_EN を “1” に設定 (読出し/書込みアクセスの暗号化が有効)
9. SMIF_DEVICEx_CTL.MERGE_TIMEOUT を “0” に設定
SMIF_DEVICEx_CTL.MERGE_EN を “1” に設定

⁸ "TX" コマンドの場合、DATA[15:8] は 2 番目に送信されるバイトを指定します。これは、DDR モードでのオクタルデータ転送、つまり DATA[17:16] = "3" および DATA[18] = "1" の場合にのみ使用されます (指定の必要あり)。

⁹ "TX_COUNT" コマンドの場合、DATA[15:0] は送信されるメモリデータユニットの数(マイナス 1)を指定します;"0": 1 ユニット、"1": 2 ユニット。SPI (DDR 付きオクタル SPI を除く)は、1 つのメモリデータユニットがバイトであり、DDR および HYPERBUS 付きオクタル SPI は 1 つのメモリデータユニットが 2 バイトワードです。使用される TX データ FIFO エントリの数 (TX_DATA_FIFO_STATUS) = 送信されるメモリデータユニットの数。

デュアルクアッド SPI DDR モードの例

10. SMIF_DEVICE_x_CTL.TOTAL_TIMEOUT を “1000” に設定
SMIF_DEVICE_x_CTL.TOTAL_TIMEOUT_EN を “1” に設定
11. SMIF_DEVICE_x_ADDR を “0x6000_0000” に設定 (デバイス領域のベースアドレスを指定)
12. SMIF_DEVICE_x_MASK を “0xFC00_0000” に設定 (デバイス領域のサイズを指定。32 M バイト 2 デバイス)
13. SMIF_DEVICE_x_ADDR_CTL.SIZE2 を “3” に設定 (XIP デバイスアドレスのサイズをバイト単位で指定: "3": 4 バイトアドレス。ReadCmd: 4QOR, WriteCmd: 4QPP)
14. SMIF_DEVICE_x_ADDR_CTL.DIV2 を “1” に設定 ('1': 2 で分割)
15. ここから 16 までは、XIP モードの読出しを設定。
SMIF_DEVICE_x_RD_CMD_CTL.CODE を “0x00” に設定 (コマンドバイトコード。連続転送では設定不要)
SMIF_DEVICE_x_RD_CMD_CTL.DDR_MODE を “0” に設定 (転送速度のモード: "0": SDR モード)
SMIF_DEVICE_x_RD_CMD_CTL.WIDTH を “0” に設定 (データ転送幅: "0": 1 ビット/サイクルの単一データ転送)
SMIF_DEVICE_x_RD_CMD_CTL.PRESENT2 を “0” に設定 (コマンドフィールドの存在: '0': 連続転送のため存在しない)
16. SMIF_DEVICE_x_RD_ADDR_CTL.DDR_MODE を “0” に設定 (転送速度のモード: "0": SDR モード)
SMIF_DEVICE_x_RD_ADDR_CTL.WIDTH を “2” に設定 (データ転送幅: "2": 4 ビット/サイクルのクアッドデータ転送)
17. SMIF_DEVICE_x_RD_MODE_CTL.CODE を “0xA5” に設定 (モードバイトコード) [S25FL256S のデータシート](#) のクアッド I/O 読出しセクションを参照。
SMIF_DEVICE_x_RD_MODE_CTL.DDR_MODE を “0” に設定 (転送速度のモード: "0": SDR モード)
SMIF_DEVICE_x_RD_MODE_CTL.WIDTH を “2” に設定 (データ転送幅: "2": 4 ビット/サイクルのクアッドデータ転送)
SMIF_DEVICE_x_RD_MODE_CTL.PRESENT2 を “1” に設定 (コマンドフィールドの存在: '1': 存在 (1 バイト))
18. SMIF_DEVICE_x_RD_DUMMY_CTL.SIZE5 を “3” に設定 (ダミーサイクルの数 (マイナス 1))。 [S25FL256S のデータシート](#) のコンフィギュレーションレジスタ 1 (CR1) のセクション参照
SMIF_DEVICE_x_RD_DUMMY_CTL.PRESENT2 を “1” に設定 (コマンドフィールドの存在: '1': 存在 (1 バイト))
19. SMIF_DEVICE_x_RD_DATA_CTL.DDR_MODE を “0” に設定 (転送速度のモード: "0": SDR モード)
SMIF_DEVICE_x_RD_DATA_CTL.WIDTH を “3” に設定 (データ転送幅: "3": 8 ビット/サイクルのオクタルデータ転送)
20. SMIF_DEVICE_x_RD_CRC_CTL.CODE を “0x0000” に設定 (読出しバス CRC 制御は未使用)
21. SMIF_DEVICE_x_RD_BOUND_CTL.CODE を “0x0000” に設定 (読出しバウンダリ制御は HYPERBUS™ に使用)
22. SMIF_DEVICE_x_WR_CRC_CTL.CODE を “0x0000” に設定 (読出しバス CRC 制御は未使用)
23. SMIF_DEVICE_x_CTL.ENABLED を “1” に設定 (デバイス有効化: '1': 有効)

デュアルクアッド SPI DDR モードの例

クアッド SPI の読出しコマンドの場合、追加のクアッド I/O 読出し命令を必要とせずにアドレスジャンプを実行できます。これは、[Figure 2](#) に示すように、アドレスシーケンスの後のモードビットの設定によって制御されます。[Table 7](#) に、クアッド I/O 読出し (4QIOR ECh) のモード設定の TX コマンド FIFO エントリを説明します。[Figure 6](#) のクアッドフラッシュの機能であるクアッド I/O ハイパフォーマンス読出しモードの詳細については、[S25FL256S のデータシート](#)を参照してください。MMIO モードでのモード設定と XIP モードでの読出しは、[Figure 2](#) に示したものと別に行われます。

Table 7 クアッド I/O 読出しの SMIF_TX_CMD_FIFO_WR 設定 (4QIOR ECh)

ビット	26:24	23	22	21	20	19	18	17:16	15:18 ^[10]	7:0	SMIF_TX_CMD_FIFO_WR
エントリ	コマンド	デバイス選択				最終 TX	SDR/DDR	データ転送幅		転送バイト	
1	0 (命令)	3				0	0	0	0	0xEC	0x030_00EC
2	0 (アドレス)	3				0	0	2	0	0x01	0x032_0001
3	0 (アドレス)	3				0	0	2	0	0x24	0x032_0024
4	0 (アドレス)	3				0	0	2	0	0x00	0x032_0000
5	0 (アドレス)	3				0	0	2	0	0x00	0x032_0000
6	0 (モード)	3				1	0	2	0	0xA5	0x032_00A5

3.7 データ学習パターン

データ学習パターン (DLP) の設定方法を以下に示します。DLP の詳細については、[\[4\]](#)を参照してください。

3.7.1 DLP の設定

DLP は初期状態で有効になっています。

3.7.2 DLP キャリブレーション

DLP キャリブレーションの設定方法について説明します。

DLP キャリブレーションハードウェアを備えたクアッド SPI デバイスは、ハードウェアによる DLP キャリブレーションを実行できます。

DLP キャリブレーションハードウェアを備えたクアッド SPI デバイスにのみ使用します。

[Figure 14](#) に、DLP キャリブレーションを示します。

¹⁰ "TX" コマンドの場合、DATA[15:8] は 2 番目に送信されたバイトを指定します。これは、DDR モードでのオクタルデータ転送、つまり DATA[17:16] = "3" および DATA[18] = "1" の場合にのみ使用されます (指定の必要あり)。

デュアルクアッド SPI DDR モードの例

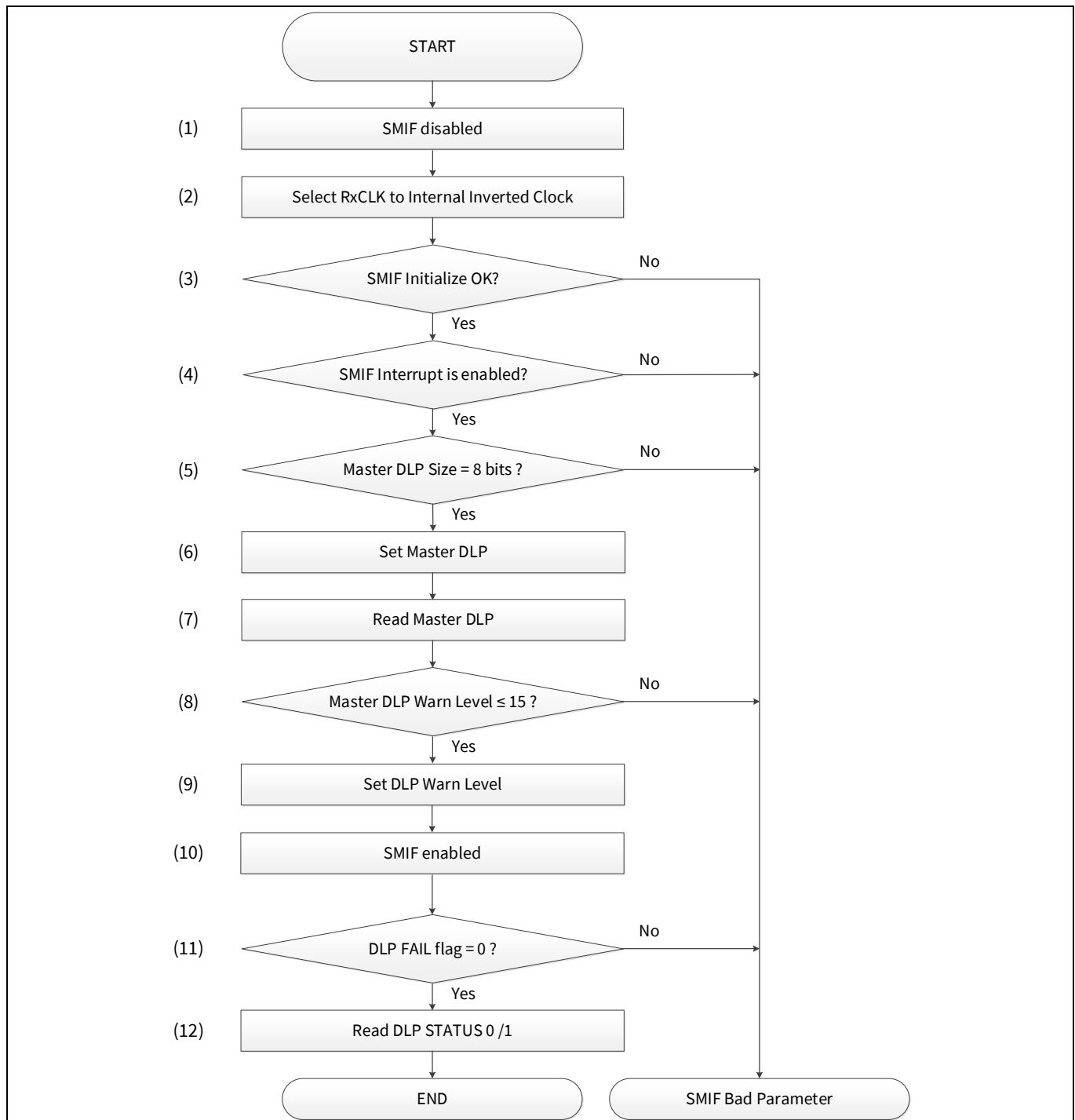


Figure 14 DLP キャリブレーション

(1) SMIF 機能を無効化

SMIF_CTL.ENABLED を“0” (SMIF は無効) に設定

(2) 内部反転クロックへ RxCLK を選択

SMIF_CTL.CLOCK_IF_RX_SEL = “5” (反転内部クロック) に設定

(3) SMIF を初期化

SMIF の初期化中に SMIF 割込みが無効になっていることを確認してください。

SMIF_INTR_MASK.TR_TX_REQ = “0” (デフォルト) に設定

SMIF_INTR_MASK.TR_RX_REQ = “0” (デフォルト) に設定

デュアルクアッド SPI DDR モードの例

- (4) SMIF の初期化が終了し、SMIF 割込みの有効化を確認
- (5) マスタ DLP のサイズを確認
- (6) 比較するマスタ DLP を設定
DLP 幅は一定の 8 ビットです。
入力値は 16 ビット整数ですが、上位 8 ビットは無視されます。
SMIF_DL_CTL.DLP = dlp を設定 (例: dlp = 0xAA)
- (7) マスタ DLP を読出し
SMIF_DL_CTL.DLP を読出し
- (8) マスタ警告レベル^[11]を確認
警告レベル (warnlevel) が 15 より大きい場合、SMIF BAD パラメーターが返されます。それ以外の場合は、SMIF Success が返されます。
SMIF_DL_CTL.DLP_WARNING_LEVEL を読出し
- (9) DLP 警告レベルを設定
SMIF_DL_CTL.DLP_WARNING_LEVEL = warnlevel を設定, (例: warnlevel = “0x3” (推奨: “0x3”以上))。
- (10) SMIF 機能を有効化
SMIF_CTL.ENABLED を“1” (SMIF は有効) に設定
- (11) 割込みフラグを確認
SMIF_INTR.DL_FAIL を読出し
値が“0”ならキャリブレーションは成功で、それ以外なら失敗です。
- (12) DLP ステータス (初期値: すべて“0”)を確認
SMIF_DL_STATUS0 を読出し
SMIF_DL_STATUS1 を読出し
ハードウェアによって書き換えられたことを確認してください。

¹¹ DLP 警告については、TRAVEO™ T2G architecture technical reference manual の“Serial Memory Interface”セクションの
“DL_WARNING”の説明を参照してください。 [関連資料](#)を参照してください。

4 ハードウェアコンフィギュレーション

4.1 デュアルクアドコンフィギュレーションでの S25FL256 デバイスの使用

この使用例は、S25FL256S QUAD SPI フラッシュメモリを使用したデュアルクアドメモリコンフィギュレーションの設定例です。

Figure 15 に、デュアルクアドメモリ接続を示します。

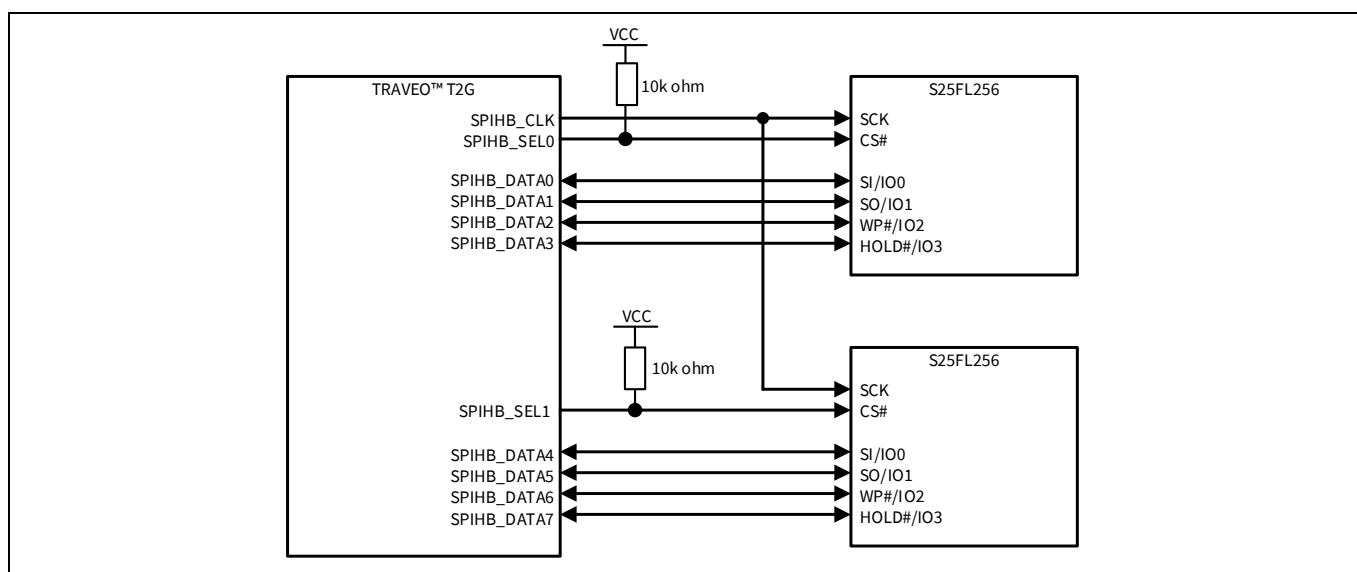


Figure 15 デュアルクアドメモリ接続

データラインに直列抵抗を追加することを推奨します。

実際の値については、システム要件に応じて 0～33 オームの範囲で適切な値を使用してください。それでもなお、お客様側でシグナルインテグリティ (SI) 解析が必要です。

4.2 HYPERFLASH™ および HYPERRAM™ コンフィギュレーションの使用

この使用例は、HYPERFLASH™ および HYPERRAM™ メモリコンフィギュレーションの設定例です。

Figure 16 に HYPERFLASH™ (S26KL512S) と HYPERRAM™ (S27KL0641) を使用した接続例を示します。

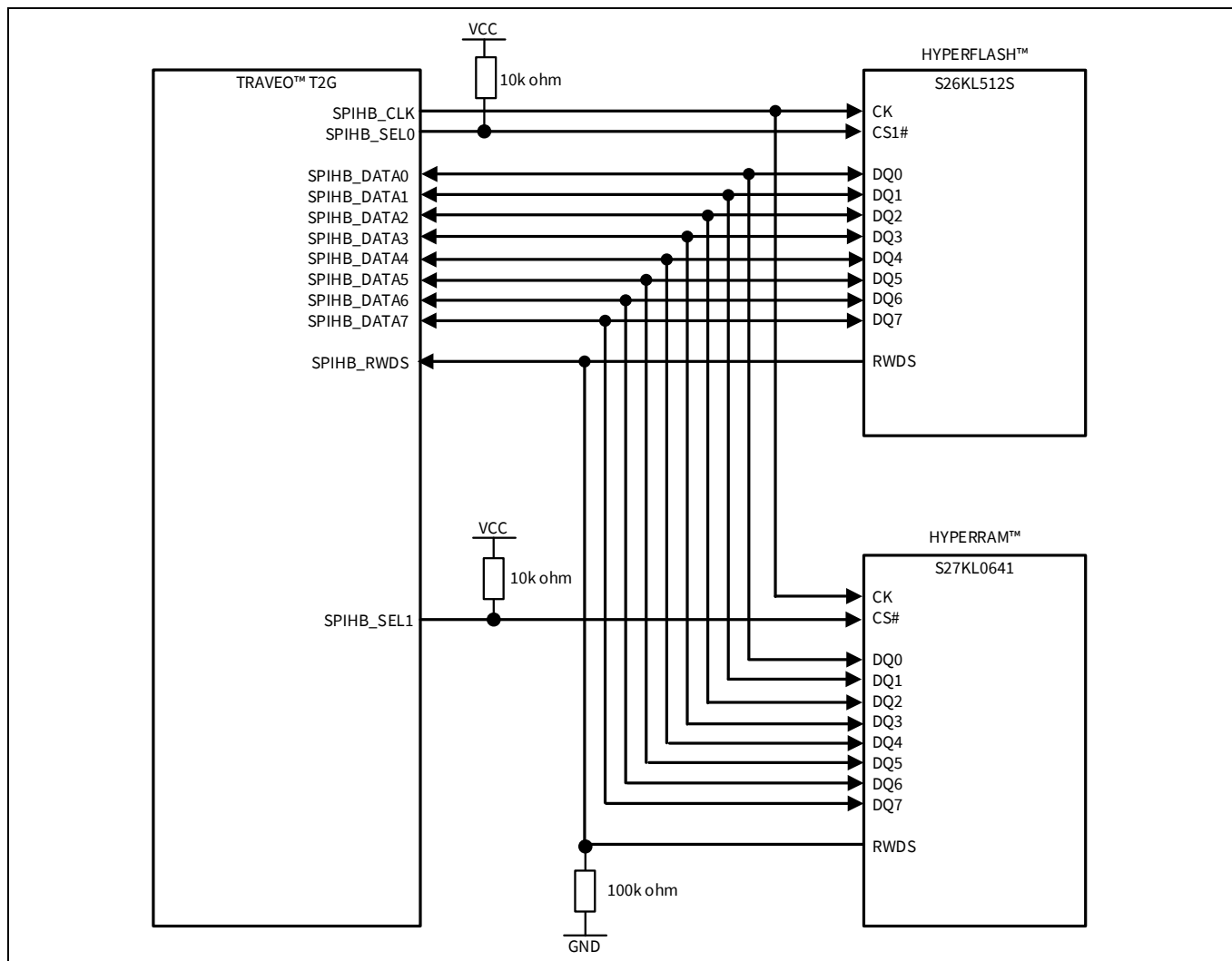


Figure 16 HYPERFLASH™ および HYPERRAM™ メモリ接続

データラインに直列抵抗を追加することを推奨します。

実際の値については、SI 分析に基づいて 0 ～ 33 オームの範囲で、システム要件に応じて適切な値を使用してください。

メモリが選択されていないときに駆動されていない RWDS 入力信号 (RX キャプチャクロックとして使用) を回避するには、RWDS 信号にプルダウンを使用する必要があります (0 はレイテンシサイクル中および最後のデータ読出し後、選択解除前の非アクティブな RWDS 状態であるため)。

この抵抗値は一例です。実際の値については、システム要件に従って適切な値を使用してください。

用語集

5 用語集

Table 8 用語集

用語	説明
AES	Advanced Encryption Standard (高度暗号化規格)
AHB	AMBA High-performance Bus (AMBA ハイパフォーマンスバス)
DDR	Dual Data Rate (デュアルデータレート)
DESELECT コマンド	TX コマンド FIFO の 5 つのコマンドタイプの 1 つ。このコマンドにより、メモリインタフェース送信ロジックが転送を終了し、メモリデバイスの選択を解除します。
DLP	Data Learning Pattern (データ学習パターン)
デュアルクアッド	2 つのデバイスがクアッド SPI メモリ。SPI 読出しおよび書込み転送中、各デバイスは専用の信号接続を使用し、1 バイトのニブルを供給する。
DUMMY_COUNT コマンド	TX コマンド FIFO の 5 つのコマンドタイプの 1 つ。このコマンドは、ダミーサイクルの数を指定します。
HYPERBUS™	低信号数 DDR インタフェース
MMIO	Memory Mapped Input/Output (メモリマップド I/O)
RX_COUNT コマンド	TX コマンド FIFO の 5 つのコマンドタイプの 1 つ。このコマンドは、データを外部メモリから RX データ FIFO に受信するときに使用されます。
SDR	Single Data Rate (シングルデータレート)
SI	Signal Integrity (シグナルインテグリティ)
SMIF	Serial Memory Interface (シリアルメモリインタフェース)
SPI	Serial Peripheral Interface (シリアルペリフェラルインタフェース)
TX コマンド	TX コマンド FIFO の 5 つのコマンドタイプの 1 つ。このコマンドは、クアッド SPI のコマンドの命令、アドレス、モードなどのフェーズを指定します。
TX_COUNT コマンド	TX コマンド FIFO の 5 つのコマンドタイプの 1 つ。このコマンドは、データが TX データ FIFO から外部メモリに送信されるときに使用します。
XIP	eXecute-In-Place (コードの直接実行)

関連資料

関連資料

以下は、TRAVEO™ T2G ファミリシリーズのデータシートとテクニカルリファレンスマニュアルです。これらのドキュメントを入手するには、[テクニカルサポート](#)に連絡してください。

[1] デバイスデータシート

- [CYT4BF datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family](#)
- CYT4DN datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family (Doc No. 002-24601)
- [CYT3BB/4BB datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family](#)
- CYT3DL datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family (Doc No. 002-27763)

[2] Body controller high ファミリ

- [TRAVEO™ T2G automotive body controller high family architecture technical reference manual \(TRM\)](#)
- [TRAVEO™ T2G automotive body controller high registers technical reference manual \(TRM\) for CYT4BF](#)
- [TRAVEO™ T2G automotive body controller high registers technical reference manual \(TRM\) for CYT3BB/4BB](#)

[3] Cluster 2D ファミリ

- TRAVEO™ T2G automotive cluster 2D family architecture technical reference manual (TRM) (Doc No. 002-25800)
- TRAVEO™ T2G automotive cluster 2D registers technical reference manual (TRM) for CYT4DN (Doc No. 002-25923)
- TRAVEO™ T2G automotive cluster 2D registers technical reference manual (TRM) for CYT3DL (Doc No. 002-29584)

[4] [S25FL128S/S25FL256S, 128 Mb \(16 MB\)/256 Mb \(32 MB\) 3.0V SPI flash memory](#)

改訂履歴

改訂履歴

版数	発行日	変更内容
**	2019-11-28	英語版 002-24454 Rev. ** の日本語版です。
*A	2020-05-27	英語版 002-24454 Rev. *A の日本語版です。
*B	2022-01-25	英語版 002-24454 Rev. *C の日本語版です。
*C	2022-08-05	英語版 002-24454 Rev. *D の日本語版です。
*D	2023-04-03	英語版 002-24454 Rev. *E の日本語版です。

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-04-03

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2023 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Go to www.infineon.com/support

Document reference

002-29072 Rev. *D

重要事項

本手引書に記載された本製品の使用に関する手引きとして提供されるものであり、いかなる場合も、本製品における特定の機能性能や品質について保証するものではありません。本製品の使用前に、当該手引書の受領者は実際の使用環境の下であらゆる本製品の機能及びその他本手引書に記載された一切の技術的情報について確認する義務が有ります。インフィニオンテクノロジーズはここに当該手引書内で記される情報につき、第三者の知的所有権の不侵害の保証を含むがこれに限らず、あらゆる種類の一切の保証および責任を否定いたします。

本文書に含まれるデータは、技術的訓練を受けた従業員のみを対象としています。本製品の対象用途への適合性、およびこれら用途に関連して本文書に記載された製品情報の完全性についての評価は、お客様の技術部門の責任にて実施してください。

警告事項

技術的要件に伴い、製品には危険物質が含まれる可能性があります。当該種別の詳細については、インフィニオンの最寄りの営業所までお問い合わせください。

インフィニオンの正式代表者が署名した書面を通じ、インフィニオンによる明示の承認が存在する場合を除き、インフィニオンの製品は、当該製品の障害またはその使用に関する一切の結果が、合理的に人的傷害を招く恐れのある一切の用途に使用することはできないこと予めご了承ください。