

## Design Guide: TIDM-1010

## BiSS-C C2000™ MCU 用の絶対エンコーダ、マスタ インターフェイスのリファレンス デザイン



## 概要

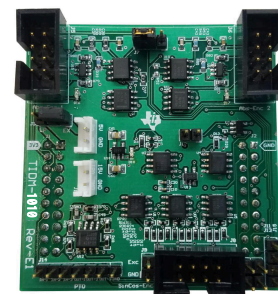
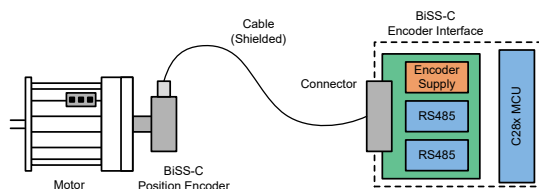
C2000™ マイクロコントローラ (MCU) の Position Manager テクノロジは、ほとんどの一般的なデジタルおよびアナログ ポジション センサと接続できる統合ソリューションで、外部の FPGA (Field Programmable Gate Array) または ASIC (Application Specific Integrated Circuit) が必要ありません。Position Manager BoosterPack™ は柔軟でコスト効果の高いプラットフォームであり、各種のエンコーダ インターフェイスの評価を目的とし、複数の C2000 MCU LaunchPad™ 開発キットと連携して動作するように設計されています。このリファレンス デザインのソフトウェアは、ポジション エンコーダ用のデジタル双方向インターフェイスである BiSS-C™ の実装を特に目的としています。このリファレンス デザインに含まれている、高度に最適化された使いやすいソフトウェア ライブラリおよびサンプルにより、Position Manager BoosterPack を使用する BiSS-C ポジション エンコーダの動作が可能になります。

## リソース

TIDM-1010	デザイン フォルダ
LAUNCHXL-F28P65X	ツール フォルダ
SN65HVD78	プロダクト フォルダ
TLV702	プロダクト フォルダ
TPS22918-Q1	プロダクト フォルダ



テキサス・インスツルメンツの E2E™ サポート エキスパートにお問い合わせください。



## 特長

- ポジション エンコーダ インターフェイス用の、柔軟で低電圧の BoosterPack 評価プラットフォーム
- 追加の FPGA を必要としない、BiSS-C 用の統合 MCU ソリューション
- ライブラリで提供されるドライバ機能およびデータ構造を使用して、BiSS-C コマンドと簡単にインターフェイス可能
- 受信したデータのパック解除、および最適化された巡回冗長性検査(CRC)アルゴリズムをライブラリでサポート
- 最高 10MHz のクロック周波数をサポートし、100m までの長さのケーブルで動作を検証済み
- BiSS-C ソフトウェア ライブラリを紹介する評価用ソフトウェア サンプルが付属

## アプリケーション

- 産業用
- モーター ドライブ

## 1 システムの説明

サーボドライブのような産業用ドライブには、高精度、高信頼性、低レイテンシの位置フィードバックが必要です。BiSS プロトコルは、センサとコントローラの間でデジタル データをシリアル転送するように設計されています。BiSS は、Bidirectional serial synchronous (双方向シリアル同期) の略です。BiSS インターフェイスは、オープンソースのプロトコルとして iC-Haus GmbH によって 2002 年に開発されました。BiSS-C モードは、BiSS-C インターフェイスが位置データを周期的に読み出す連続モードです。

TIDM-1010 設計には、C2000 LaunchPad に対する BiSS-C インターフェイスが実装されています。BiSS-C は、RS-485 規格に基づいた、純粋なシリアルのデジタル インターフェイスです。BiSS-C は、位置の値だけでなく他の物理量を送信でき、エンコーダの内部メモリの読み書きも可能です。送信されるデータタイプには、絶対位置、回転数、温度、パラメータ、診断があります。図 1-1 に、この設計で使用される主要なハードウェア ブロックを示します。

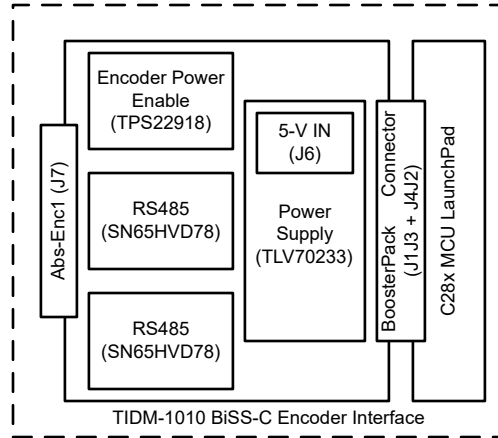


図 1-1. TIDM-1010 のハードウェア ブロックおよびコネクタ

TIDM-1010 は、ポイント ツー ポイント構成をサポートしており、通常は BiSS 位置エンコーダまたはロータリー エンコーダとともに使用されます。ポイント ツー ポイントトポロジを 図 1-2 に示します。ポイント ツー ポイント構成では、1 つまたは複数のセンサーを備えた 1 つのデバイスのみがマスタによって動作します。PM\_bissc ライブラリは、この構成のみをサポートします。

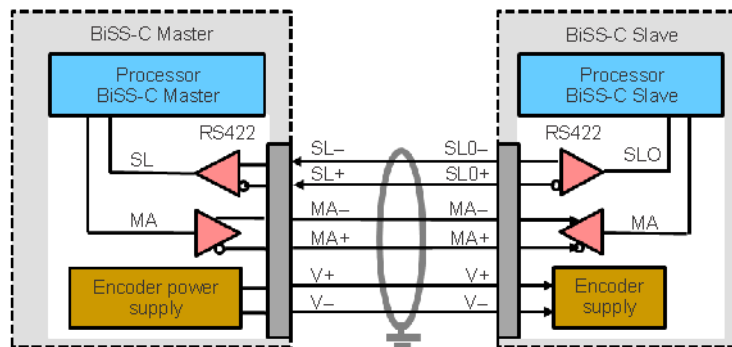


図 1-2. BiSS-C ポイント ツー ポイント構成

絶対位置エンコーダは、6 芯シールド ケーブルで TIDM-1010 デバイスに接続されます。物理層として RS-485 を使用しています。6 本の電線は次のとおりです。

- MA+ および MA-: エンコーダ クロック (BiSS マスタ クロック) の差動信号
- SL+ および SL-: エンコーダから返されるデータ (BiSS スレーブ リターン) の差動信号
- V+ および V-: エンコーダの電源およびグランド

BiSS-C MA クロックの周波数は可変で、ケーブル長によって異なります (セクション 3.3.4 を参照)。エンコーダおよびエンコーダ ケーブルに応じて、最大ケーブル長または実現可能な最大クロック周波数は異なる場合があります。ケーブルの

品質は通信性能に影響するため、エンコーダのメーカーはデータシートでこれらの制限を定義し、エンコーダでの使用に適したケーブルを推奨しています。プロトコルおよびポイント ツー ポイント構成の詳細については、[BiSS インターフェイス: BiSS ユーザー ソサエティおよびインターネット プラットフォーム](#)をご覧ください。

テキサス・インスツルメンツの C2000 Position Manager BiSS-C (PM\_bissc) エンコーダ インターフェイスを使用すると、FPGA や CPLD のような外部ハードウェアを使用せずに BiSS を実装できます。このリファレンス実装には、次の特長があります。

- 10m のケーブル長で最大 10MHz のクロック周波数に対応
- ケーブル伝搬遅延補償を内蔵
- ソフトウェアドライバの機能
  - エンコーダでトランザクションを実行します。MA 信号をエンコーダに送信し、応答を受信します。
  - CRC を計算します。
  - 受信した CRC を、計算された CRC と比較します。
  - 応答データを展開します

このリファレンス実装には、すべてのソースコードが含まれています。ユーザーは、アプリケーションの必要に応じて、必要な変更を実装に加えることができます。

**注**

このライブラリは、以下に示す基本インターフェイスドライバをサポートしています。

- シングル サイクル データ (位置 + エラー + 警告 + CRC) トランザクション
- シングル レジスタ読み取りアクセス
- シングル レジスタ書き込みアクセス

上位レベルのアプリケーション ソフトウェアおよび BiSS 機能はすべて、この実装で提供されている基本インターフェイスを使用して、ユーザーが開発することができます。

**1.1 主なシステム仕様**

**表 1-1. 主なシステム仕様**

パラメータ	仕様	詳細
入力電圧	5V <sup>(1)</sup>	<a href="#">セクション 3.3.1</a>
出力電圧 (エンコーダ)	5 V	<a href="#">セクション 3.3.1</a>
サポートされているプロトコル	BiSS-C ポイント ツー ポイント	<a href="#">BiSS</a>
最高周波数	10 MHz	最大 10m のケーブルをサポート。 <a href="#">セクション 2.3.1.1</a>
エンコーダのビット数	BiSS-C プロトコル標準	<a href="#">BiSS</a>
位置データの CRC	$x^6 + x + 1$	位置データ (シングル サイクル データ) 検証の多項式
制御データの CRC	$x^4 + x + 1$	制御データ検証の多項式
制御データの機能	シングル レジスタ読み取り、シングル レジスタ書き込み	
CPU のサイクル数	<a href="#">セクション 3.3.5</a>	
コードのサイズ	<a href="#">セクション 3.3.5</a>	

(1) この電源に必要な電流制限は、TIDM-1010 デバイスに接続されているエンコーダによって決まります。TIDM-1010 が生成する電圧ではなく、調整可能な電流制限付きの汎用ベンチトップ型可変電源を使用することを推奨します。

## 2 システム概要

C2000 BiSS TIDM-1010 リファレンス デザインは、ハードウェアとソフトウェアの組み合わせです。コアのハードウェア コンポーネントは、C2000 リアルタイム マイクロコントローラ (MCU) と RS-485 トランシーバです。この実装で使用する基板は、C2000 LaunchPad と、RS-485 トランシーバを搭載した TIDM-1010 boosterPack です。C2000Ware モーター制御 SDK パッケージには、必要なソフトウェアが含まれています。このソフトウェアには、主要な BiSS インターフェイス機能を実装するライブラリと、BiSS 通信のデモを行うためのシステムレベルのサンプルが含まれています。

BiSS エンコーダ インターフェイスは、C2000 CLB (構成可能ロジックブロック) と SPI (シリアル ペリフェラル インターフェイス) モジュールを活用しています。CLB は、MA クロック、SPI クロックを制御し、ケーブルの伝搬遅延を補償します。SPI モジュールは、RS-485 物理層に対する受信インターフェイスとして機能します。ファームウェアは C 言語で記述されており、C2000 MCU の C28x 上で動作します。

C2000 LaunchPad は、TIDM-1010 RS-485 トランシーバに電力を供給し、エンコーダに 5V を供給できます。エンコーダの仕様で、LaunchPad が供給できるよりも多くの電流が必要な場合は、別に 5V を供給することもできます。

スタートアップ時に、C28x で動作するアプリケーションによって MCU クロックが初期化され、ピンの多重化が設定されます。MCU の SPI および CLB も、データの送受信の必要に応じて構成されます。

### 2.1 ブロック図

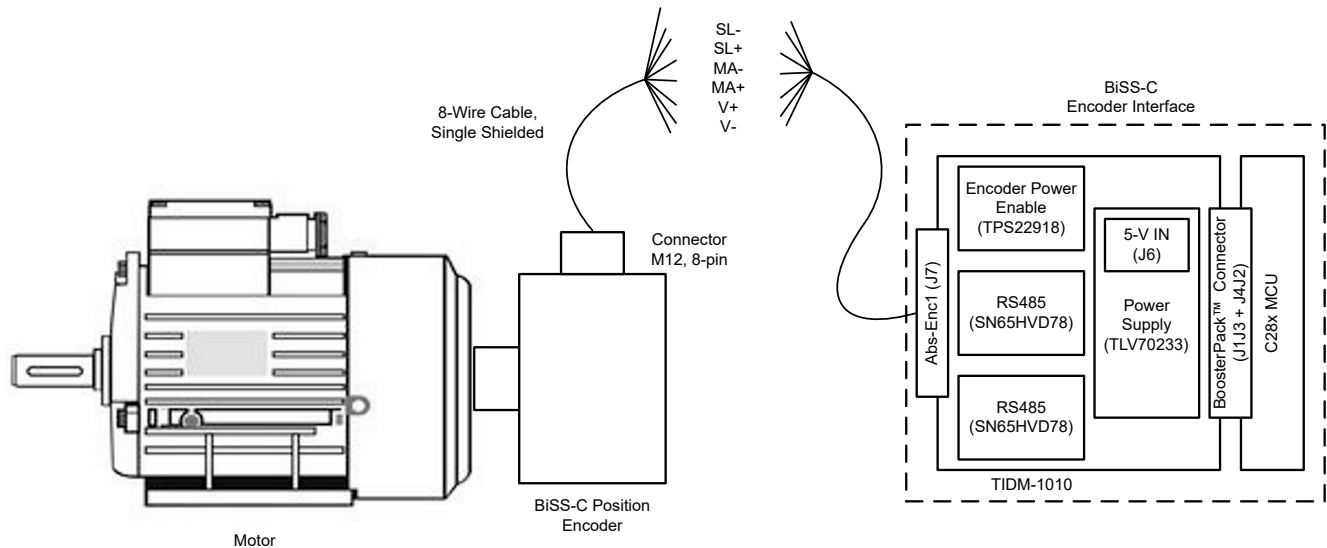


図 2-1. TIDM-1010 システム ブロック図

### 2.2 主な使用製品

TIDM-1011 リファレンス デザインのハードウェアは、C2000 LaunchPad と BOOSTXL-POSMGR BoosterPack で構成されています。このセクションでは、使用されている主なデバイスについて説明します。これらの各デバイスの詳細については、TI.com で各プロダクト フォルダをご覧ください。

#### 2.2.1 C2000 リアルタイム MCU LaunchPad

提供される TIDM-1010 サンプルは、TMS320F28P650DK9 MCU をベースとしています。F28P65x は、200MHz のデュアル C28x CPU と、リアルタイム制御コプロセッサ (CLA) で、600 MIPS の総合システム性能を実現しています。この強力な MCU は 1MB のオンボード フラッシュを搭載し、16 ビットまたは 12 ビット A/D コンバータ (ADC)、コンパレータ、12 ビット D/A コンバータ (DAC)、デルタ シグマ sinc フィルタ、HRPWM、eCAP、eQEP、CAN など、高度な差別化を可能にするペリフェラルを内蔵しています。

F28P65x には、構成可能ロジックブロック (CLB) も搭載されています。BiSS インターフェイスにより、CLB を広範に活用できます。CLB ペリフェラルを利用すれば、ユーザーは外部 FPGA や CLPD を必要とせずにカスタム ロジックを組み込むことができます。CLB は複数のサブモジュールで構成されており、これらを組み合わせてカスタムのデジタル ロジックを

実現します。サブモジュールには、有限ステートマシン (FSM)、ルックアップ テーブル (LUT)、カウンタがあります。また、CLB は既存のオンチップ制御ペリフェラルと接続して機能を強化し、設計の選択肢として使用できます。BiSS インターフェイスは、CLB を搭載した任意の C2000 デバイスに移植できます。

LAUNCHXL-F28P65X 評価ボードを利用して、TIDM-1010 を実装できます。LAUNCHXL-F28P65X LaunchPad は、F28P65x デバイス向けの低コスト開発キットです。

### 2.2.2 SN65HVD78

SN65HVD78 デバイスは差動ドライバと差動レシーバを組み合わせたもので、どちらも 3.3V の単一電源で動作します。ドライバの差動出力とレシーバの差動入力はい内部的に接続され、半二重 (2 線式バス) 通信に適したバスポートを形成しています。これらのデバイスは同相電圧範囲が広いいため、長いケーブルを使用するマルチポイントアプリケーションに適しています。

デバイスの全機能と仕様については、[SN65HVD78](#) のプロダクトフォルダを参照してください。

### 2.2.3 TLV702

TLV702 シリーズの低ドロップアウト (LDO) リニアレギュレータは、静止電流が小さく、ラインおよび負荷過渡性能が非常に優れています。どのバージョンのデバイスも、安全のためにサーマルシャットダウン機能と電流制限機能を搭載しています。これらのデバイスは、出力負荷なしでも指定の精度へのレギュレーションを行います。

デバイスの全機能と仕様については、[TLV702](#) のプロダクトフォルダを参照してください。

### 2.2.4 TPS22918-Q1

TPS22918-Q1 は、シングルチャネル負荷スイッチで、立ち上がり時間とクイック出力放電機能の両方を構成可能です。このデバイスには、最大 2A の連続電流をサポートできる N チャネル MOSFET が内蔵されています。スイッチはオン/オフ入力で制御され、低電圧の制御信号と直接接続できます。

デバイスの全機能と仕様については、[TPS22918-Q1](#) のプロダクトフォルダを参照してください。

## 2.3 設計上の考慮事項

このセクションでは、次の説明を行います。

1. BiSS-C インターフェイス プロトコルの概要。
2. C2000 BiSS-C エンコーダ インターフェイスの概要。
3. TIDM-1010 ハードウェア (BOOSTXL-POSMGR BoosterPack) の実装。
4. 必要な入出力、CRC の計算、CLB の設計を含めた、C2000 MCU の実装。
5. C2000 BiSS-C エンコーダ インターフェイス ソフトウェア ライブラリの概要。

### 注

このセクションで説明するのは、実装の詳細のみです。次のような関連情報については、記載の資料を参照してください。

- ハードウェア要件、セットアップ、テスト: [セクション 3](#) を参照してください。
- ソフトウェア: ソフトウェアのインストールと実行: 下記を参照してください。『C2000 BiSS-C エンコーダ インターフェイスのソフトウェアガイド』([html](#)、[pdf](#))ソフトウェアガイドには、次の資料が含まれています。
  - 通信のデモ
  - BiSS-C アプリケーション プログラマ インターフェイス (API)
  - 独自のソリューションにライブラリを組み込む
  - デュアルコア デバイス上の CPU2 へのライブラリの移植

### 2.3.1 BiSS-C プロトコル

BiSS-C インターフェイス仕様は、エンコーダや他のセンサと通信するためのシリアル インターフェイス プロトコルを記述しています。BiSS-C では、位置データと制御データを同一回線上で同時に送信することができます。このインターフェイスは、シリアル同期インターフェイス (SSI) プロトコルに似ており、データ送信がコントローラのクロック信号と同期していません。



表 2-1. SSI と BiSS-C の比較

SSI (シリアル同期インターフェイス)	BiSS-C
2Mbits/s	10m ケーブルで最大 10Mbit/s
該当なし	ケーブル長伝搬補償
差動ツイスト ペア ケーブル (RS-422/485)	差動ツイスト ペア ケーブル (RS-422/485)
単方向。エンコーダからコントローラへのデータ送信のみをサポート。	SSI のように単方向に使用されることが多い。双方向データ伝送もサポートしているため、ドライブによってエンコーダ構成を制御可能。
単純なパリティ チェックをサポート	巡回冗長検査 (CRC) による堅牢なエラー チェック
該当なし	エンコーダが追加処理時間を要求可能
ポイントツー ポイントトポロジ	通常、ポイントツー ポイントだが、このプロトコルはデジター チェーン接続エンコーダもサポート。このドキュメントで説明する C2000 設計では、ポイントツー ポイントのトポロジを実装します。

## 注

このドキュメントでは、BiSS-C フレームおよび制御通信のいくつかの重要な概念について説明します。詳細な仕様については、[BiSS ユーザー ソサエティ](#)および[インターネット プラットフォーム](#)を参照してください

通常、位置エンコーダは、モーターのフィードバック位置データを提供しますが、このエンコーダではモーターの閉ループ制御が可能です。このデザインでは、ポイントツー ポイント通信用の BiSS-C インターフェイスを実装しています。ポイントツー ポイント構成では、1 つのエンコーダのみがコントローラまたはドライブに接続されます。

BiSS-C プロトコルでは、コントローラがエンコーダにクロック (MA) を供給し、エンコーダはデータをコントローラ (SL) に送り返します。通信データは、シングル サイクル データ (SCD) と制御データ (CD) で構成されています。

- SCD は、エンコーダからコントローラに送信される主要なデータであり、BiSS サイクルごとに更新されます。SCD には、絶対位置、エラー ビット、警告ビット、および CRC チェックが含まれます。絶対位置には、シングルターンのみ (1 回転)、またはシングルターンとマルチターンの両方 (回転数) のデータを含めることができます。
- CD により、コントローラは、エンコーダレジスタの書き込みと読み取りを行うことができます。サイクルごとに、コントローラは、1 つの制御データビット (CDM) をエンコーダに送信し、エンコーダは、1 つの制御データビット (CDS) をコントローラに返します。

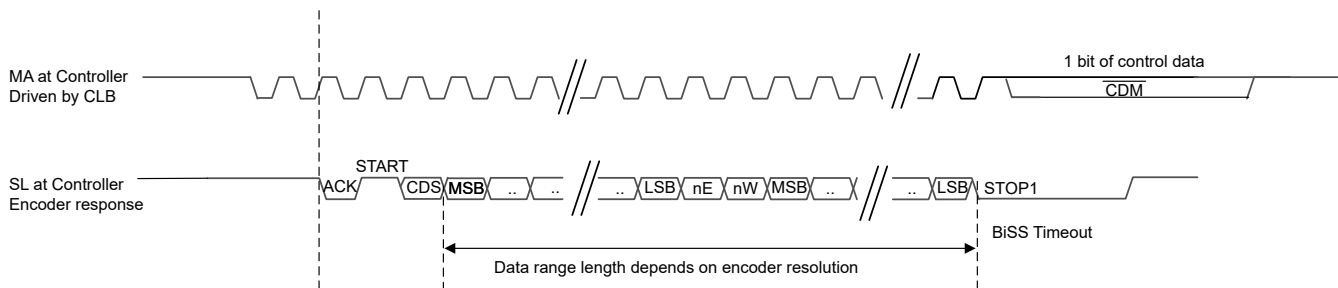


図 2-2. BiSS-C フレーム

図 2-2 に、BiSS フレームの例を示します。リセット状態では、MA ラインと SL ラインの両方がアクティブ High です。コントローラは、MA ライン経由でクロックを送信してフレームを開始します。MA クロックの 2 番目の立ち上がりエッジで、エンコーダは Low 信号で応答して BiSS フレームをアックノリッジ (ACK) します。次の MA クロック サイクルでは、エンコーダによってスタートビットがアサートされます。スタートの次に、エンコーダは、前のフレームで送信された制御 CDM ビットへの応答となる制御データ (CDS) ビットを送信します。

CDS ビットの次に、エンコーダは、位置データを最上位ビット (MSB) から順に送信します。位置データの次に、エラービット (nE) および警告ビット (nW) が続きます。次に、エンコーダは巡回冗長検査 (CRC) ビットを MSB から順に送信します。CRC は、エンコーダによって反転されて SL ライン経由で送信されます。

すべてのビットを送信すると、エンコーダは BiSS タイムアウト状態になり、SL を Low 信号レベルに駆動します。その後、エンコーダで次の送信準備ができると、または BiSS タイムアウトが満了すると、SL が High になります。BiSS タイムアウト

中の MA クロックラインの反転状態は、制御通信のための CDM ビットになります。各 BiSS フレームで、コントローラから 1 つの制御データビットが送信され、エンコーダから 1 つのビットが返されます。

注

エンコーダには、SLI (入力) 信号と SLO (出力) 信号があります。ポイントツーポイント構成では、SLO 信号はコントローラの SL に直接接続されます。

2.3.1.1 ライン遅延補償

実際のアプリケーション環境では、エンコーダはコントローラから遠く離れた場所に存在する可能性があります。エンコーダとコントローラの間には長いケーブルを接続すると、送信遅延や物理的ノイズが発生する可能性があります。

ライン遅延は、送信に使用されるケーブルの長さによる伝搬遅延です。コントローラが MA クロックの送信を開始したとき、クロックがエンコーダに到達するまでにある程度の時間が必要です。エンコーダがクロックを受信すると、エンコーダは SL データにより応答を開始します。エンコーダの応答も、ケーブルを通過してコントローラに向かって逆方向の経路を進みます。電線を経由するデータ送信における遅延時間は、ケーブルの長さ按比例します。ケーブル長が 100m までの場合、コントローラがクロックを送信してから、コントローラがエンコーダの応答を受信するまでに、1µs のケーブル遅延が発生する可能性があります。

BiSS-C インターフェイスには、ライン遅延を補償して、長いケーブルの伝送でのエラーを回避するメカニズムがあります。

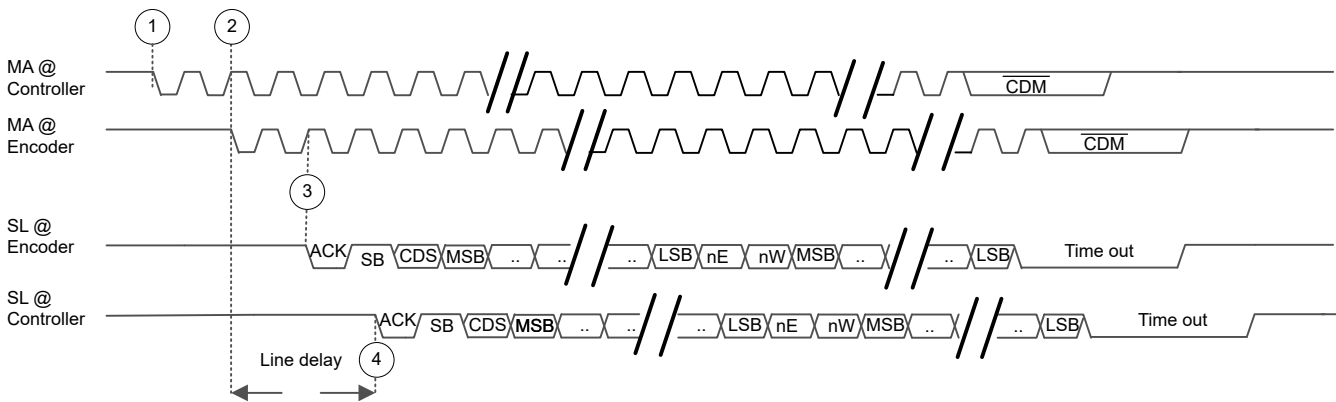


図 2-3. BiSS-C ライン遅延

図 2-3 に、コントローラ側とエンコーダ側の 2 つの観点からの信号を示します。

図 2-3 のマーカー (1) を参照:

コントローララインの MA は、クロックが BiSS-C インターフェイスでどのように見えるかを示しています。これは、ドライブコントローラが配置されているコールド側です。コントローラは MA 信号を送信してトランザクションを開始します。

図 2-3 のマーカー (2) を参照:

ライン遅延のため、MA クロック信号は、エンコーダ (モーター) で見ると遅延しています。[MA at Encoder] の行は、その遅延を示しています。

図 2-3 のマーカー (3) を参照:

エンコーダは遅延した MA クロックの 2 番目の立ち上がりエッジに応答します。[SLO at Encoder] の行は、[MA at Encoder] に対するエンコーダの応答を示しています。

図 2-3 のマーカー (4) を参照:

この応答がコントローラに戻るには、ある程度の時間がかかります。コントローラへの信号伝達は、[SLO at Controller] 信号に示すように遅延します。MA クロックの 2 番目の立ち上がりエッジから SLO ラインの最初の立ち下がりエッジまでの時間を測定することで、遅延時間の合計を計算できます。送信エラーを防止するために、BiSS-C インターフェイスはこのライン遅延を補償します。

### 2.3.1.2 エンコーダによる処理時間要求

エンコーダは、センサ データを送信する前に処理時間を要求できます。A/D 変換やメモリ アクセスなどの動作には、追加の時間が必要です。エンコーダは、スタートビット (SB) を遅延させることで処理時間を示します。コントローラは、エンコーダが処理時間を要求しているかどうかを確認し、追加のクロック サイクルを提供する必要があります。

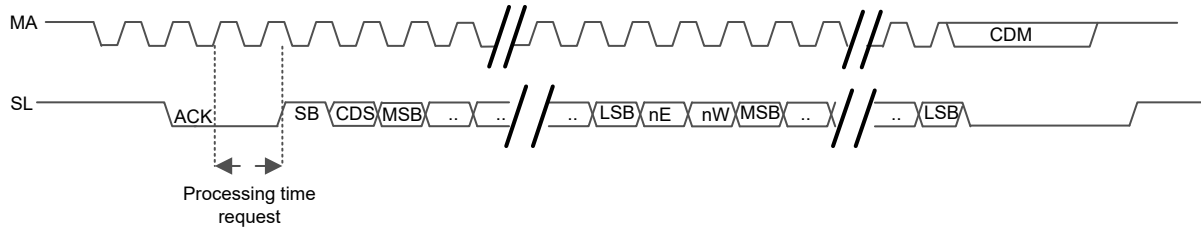


図 2-4. エンコーダの処理時間要求

### 2.3.1.3 制御通信

BiSS-C 通信では、コントローラは、位置データ通信を中断せずに MA ラインを介して制御フレームを送信できます。これは、各 BiSS フレーム内に 1 ビットの制御フレームを送信することで実現されます。

セクション 2.3.1 で説明しているように、コントローラは、BiSS フレームごとに CDM と呼ばれる 1 つのデータビットを送信します。同様に、エンコーダは、これらの CDM ビットに対して、BiSS フレームごとに CDS と呼ばれる 1 ビットによって応答します。これは、複数の BiSS フレームにわたって、制御フレーム全体が送信され、応答が受信されるまで繰り返されます。

BiSS-C 制御フレームには、次の 2 種類があります。

- レジスタ通信フレーム: エンコーダの内部レジスタの読み取りまたは書き込み
- コマンド フレーム: エンコーダへコマンドを送信

#### 注

提供された状態では、TIDM-1010 はコマンド フレームを実装していません。この機能は、システム開発者が必要に応じて、設計に追加することができます。コマンド フレームで、制御選択ビット (CTS) は 0 です (CTS = 0)。コマンド フレームを使用してマルチポイント接続をサポートする方法については、この設計では説明しません。したがって、このドキュメントでは、レジスタ通信フレームのみに集中して説明します。

読み取りまたは書き込みアクセスは、以下に示す手順によって行います。図 2-5 および図 2-6 を参照してください。

- コントローラは、CDM = 0 で 14 個以上の BiSS-C フレームを送信します
- CDM = 1 は、制御フレームのスタートビット S を示します。
- その次の CDM ビットは、CTS (制御選択ビット) と呼ばれます。レジスタ アクセスの場合、CTS は 1 です。
- 次に、コントローラは 3 ビットの ID を送信して、アクセス対象のスレーブを識別します。
- ID の後に、7 ビットのレジスタ アドレスと CRC が続きます。
- 次の 3 ビットは、読み出しビット (R)、書き込みビット (W)、スタートビット (S) です。R W S は次のように定義されます。
  - 書き込みアクセス: RWS = 011b
  - 読み出しアクセス: RWS = 101b
- コントローラは次のいずれかを行います。
  - 読み取りアクセスの場合、CDM ビットを Low に保持します
  - 書き込みアクセスの場合、書き込み対象の 8 ビット データ + CRC を送信します
- ストップ ビット (P) は、制御フレームの終了を示します。



注

ステップ 6 において、このプロトコルでは、エンコーダは、読み取りまたは書き込みのために追加の処理時間を要求できます。これは、(図に示されている  $S = 1$  ではなく)  $S = 0$  で応答することで行われます。これは現在の実装ではサポートされていませんが、開発者は CD ステートマシンの C コードを更新して追加することができます。この余分な時間は、すべてのエンコーダで必要となるわけではありません。使用する特定のエンコーダの仕様を参照してください。

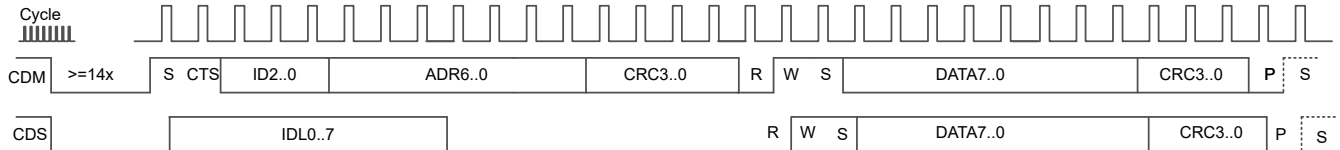


図 2-5. 制御フレーム:レジスタ読み出し

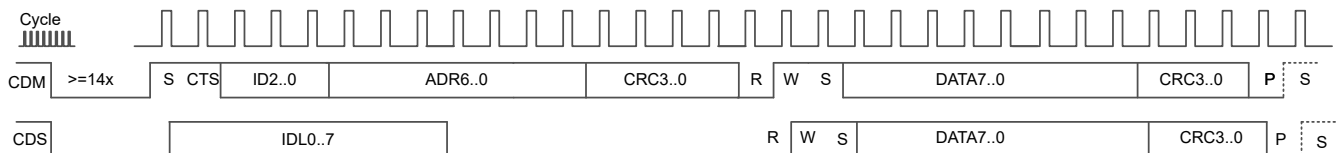


図 2-6. 制御フレーム:レジスタ書き込み

注

BiSS プロトコルでは、連続するレジスタに対して、連続読み取り、または連続書き込みが可能です。これは、コントローラがストップビット ( $P = 1$ ) を送信し、その直後に別のスタートビット ( $S = 1$ ) を送信するとイネーブルになります。この機能は TIDM-1010 には実装されていません。コントロールフレームごとに 1 回の読み取りまたは 1 回の書き込みのみがサポートされます。

### 2.3.2 C2000 BiSS-C エンコーダ インターフェイスの概要

BiSS-C エンコーダ インターフェイス上の通信は、主に次のコンポーネントによって行われます。

- CPU (C28x)
  - デバイス、CLB、SPI を構成する
  - CLB カウンタを初期化して、エンコーダの分解能に適した MA クロック周波数およびクロック数を生成する
  - データをパックおよび展開する
  - シングル サイクル データの CRC および制御フレームの CRC を計算する
  - 計算された CRC を受信した CRC と比較する
- 構成可能ロジック ブロック (CLB)
  - MA クロックおよび CDM ビットを送信する
  - SPI PICO 信号でエンコーダの応答を監視する。応答を読み取るための SPI クロックを制御する
  - インターフェイスの要求に応じて、ケーブル伝搬遅延を測定および補償する
- シリアル ペリフェラル インターフェイス (SPI)
  - エンコーダの応答を受信する
- デバイスの相互接続 (XBAR、CLB XBAR)
  - CLB とデバイスの間で信号をルーティングする
- 外部インターフェイス ブロック
  - RS-485 差動ラインドライバを搭載した TIDM-1010 基板

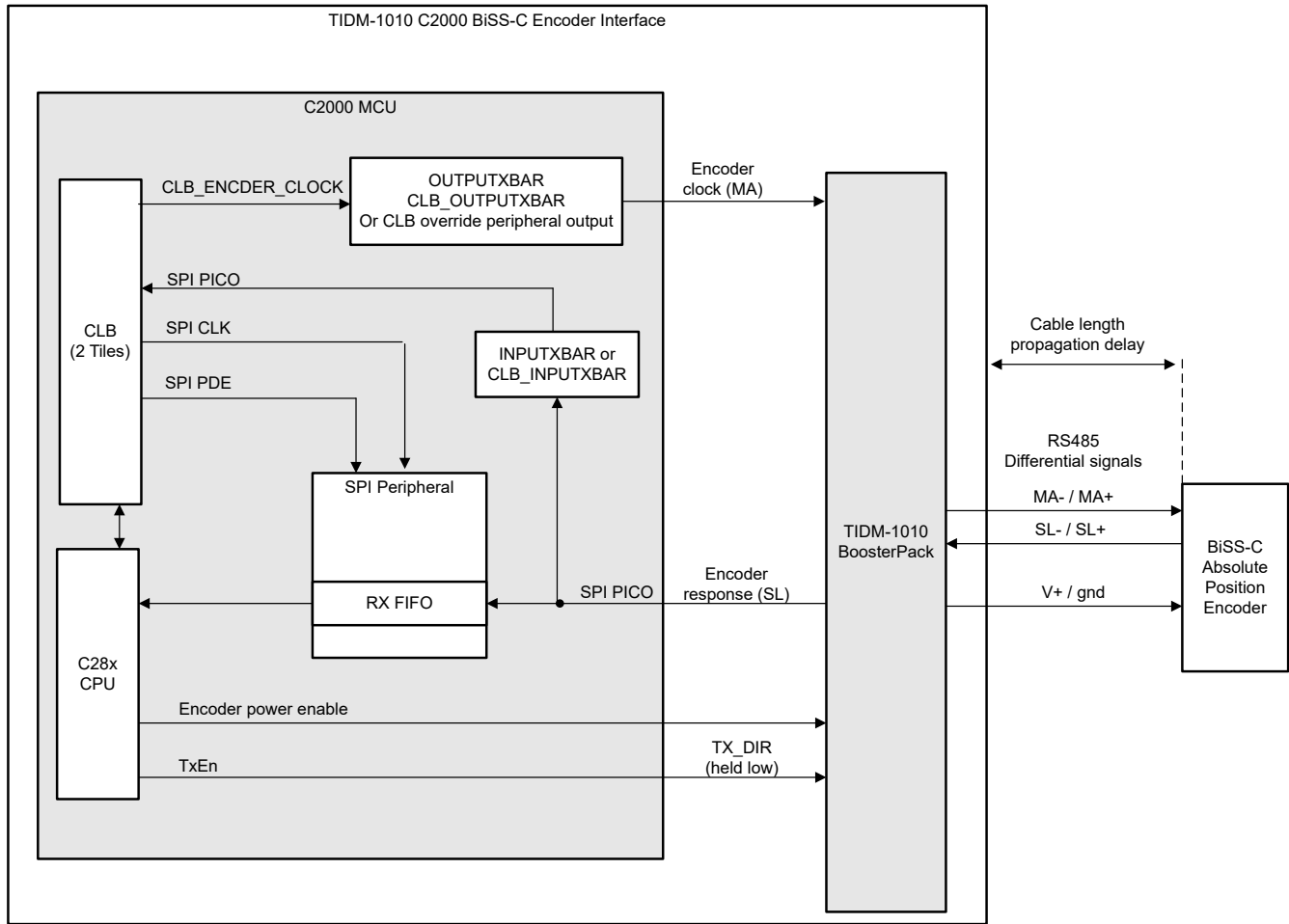


図 2-7. エンコーダ インターフェイスの実装ブロック図

注

F2837xD / F2837xS / F28007 デバイスでは、CLB は SPI 入力信号を直接オーバーライドできません。TIDM-1010 ハードウェアでは、CLB で生成された SPI クロックをペリフェラル クロック入力ピンに配線して、SPI PDE ピンをグラウンドに接続できます。詳細については、TIDM-1010 回路図を参照してください。

このセクションの残りの部分では、この設計の以下の側面について説明します。

- TIDM-1010 ハードウェア
- CLB を含む C2000 MCU のリソース
- エンコーダ インターフェイスで使用される C2000 ソフトウェア

### 2.3.3 TIDM-1010 の基板実装

TIDM-1010 ボードには次の機能があります。

- C2000 MCU とエンコーダの間で RS-485 通信を行うための差動ラインドライバおよびレシーバ。
- MCU から RS-485 ドライバ / レシーバの方向制御に送信される TxEN 信号。BiSS-C 実装では、この信号は Low に保持されます。
- SPICLK 信号は GPIO に送られ、CLB ペリフェラルから制御できます。この接続は、F2837xD、F2837xS、F28007x デバイスを除いて、サポートされているすべてのデバイスで使用できるオプションです。他のデバイスでは、CLB は、デバイス内の SPI ペリフェラルにクロックを供給できます。

注

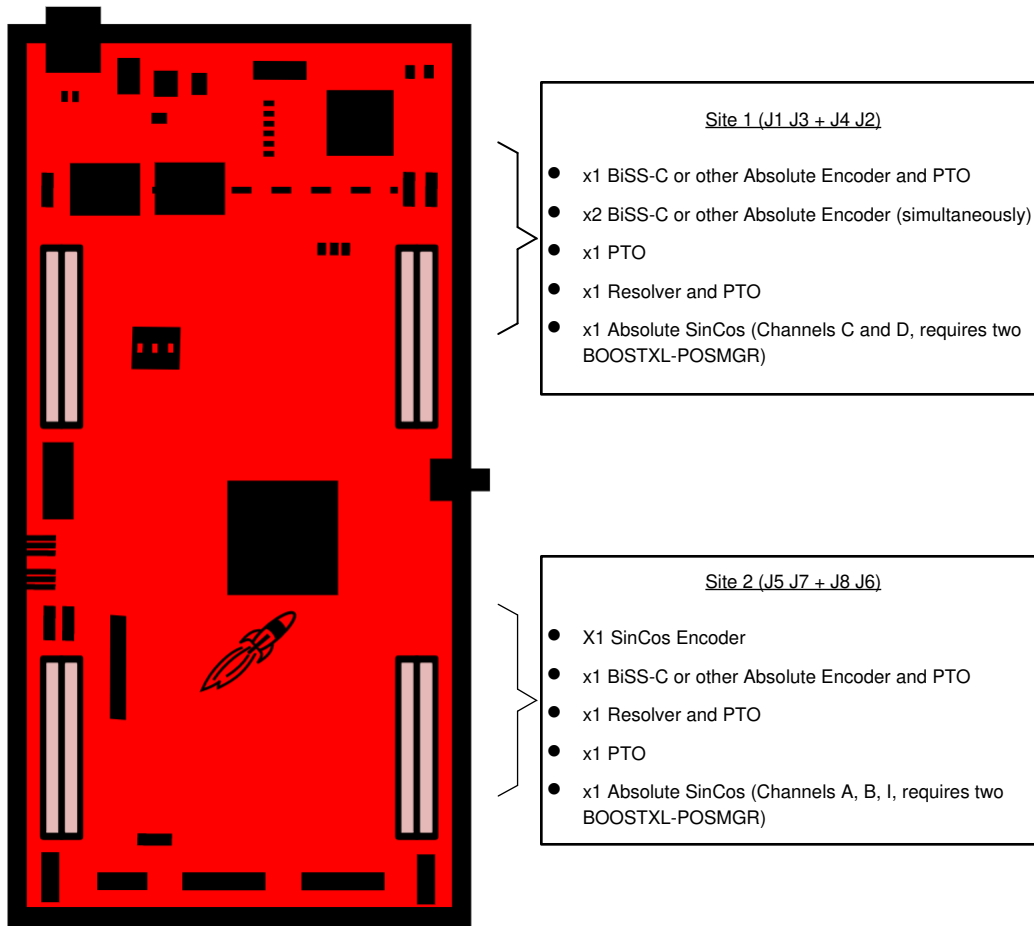
TIDM-1010 基板は Position Manager BoosterPack (BOOSTXL-POSMGR) と同じなので、TIDM-1010 は複数の種類の他のポジション エンコーダと接続できます。この基板は、将来の互換性を確保するために、デフォルトで完全に実装されています。このリファレンス デザインは BiSS-C に特化しており、このドキュメントに記載されていないハードウェア ブロックは無視できます。

TIDM-1010 BiSS-C に実装されているコネクタおよびこれらのコネクタの機能を 表 2-2 に示します。

**表 2-2. TIDM-1010 基板と BOOSTXL-POSMGR コネクタ**

コネクタ	説明	TIDM-1010 で使用
Abs-Enc-1 (J7)	BiSS-C および他の絶対エンコーダ	はい、LaunchPad サイト 2
Abs-Enc-2 (J8)	BiSS-C および他の絶対エンコーダ	なし
Abs-Enc-2 Breakout (J10)	ジャンパを使用して、サイト 2 に 2 個の絶対エンコーダを配置	なし
SinCos (J14)	SinCos エンコーダ	なし
リゾルバ (J14 と J15)	15V 励起回路を搭載したリゾルバ インターフェイス	なし
PTO (J17)	パルストレイン出力	なし
J1、J3 と J4、J2	BoosterPack コネクタ	あり
J6	5V DC 電源入力	あり
J16	15V DC リゾルバ励起入力	なし

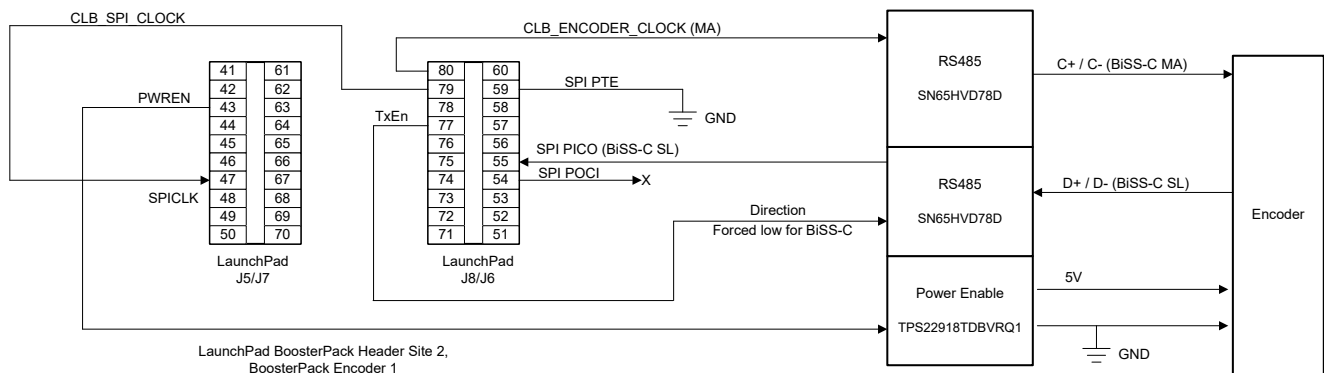
LaunchPad 開発キットの各サイトでのエンコーダ サポートを、[図 2-8](#) に示します。



**図 2-8. TIDM-1010 基板と BOOSTXL-POSMGR エンコーダのサポート**

提供された状態では、TIDM-1010 は LaunchPad サイト 2 と BOOSTXL-POSMGR のエンコーダ 1 接続を使用します。接続を、[図 2-9](#) に示します。TIDM-1010 BoosterPack の完全な回路図は、BOOSTXL-POSMGR プロダクト ページからダウンロードできます。

CLB\_SPI\_CLK to SPICLK connection can be internal to the MCU on all devices except F2837xD, F2837xS, and F28007x



**図 2-9. BoosterPack のブロック図**

### 2.3.4 MCU のリソース要件

TIDM-1010 リファレンス デザインで使用される C2000 リアルタイム MCU のリソースを、表 2-3 に示します。

表 2-3. TIDM-1010 のリソースの使用

リソース名と数量	タイプ	目的
CLB × 2	タイプ 1 またはそれ以降	SPI クロック、遅延補償、CDM ビット制御を行います。タイル インスタンスを変更した場合、CLB の入出力配線も更新する必要があります。
GPIO × 3	I/O	<ul style="list-style-type: none"> <li>CLB 出力、MA エンコーダ クロック</li> <li>CLB 出力、RS-485 方向制御 (TxEN)。BiSS-C の場合、この信号は Low に保持されます。</li> <li>CPU エンコーダの電源制御 (PwrCtl)エンコーダの電源を別の方法で制御する場合 (例:外部電源)、この信号は必要ありません。</li> </ul>
GPIO × 1	I/O (F2837xD、F2837xS、F28007xのみ)	<ul style="list-style-type: none"> <li>CLB_SPI_CLK の CLB 出力</li> <li>CLB タイプ 1:このピンは SPICLK 入力の外部に配線します。</li> <li>CLB タイプ 2 またはそれ以降:CLB から SPI モジュールに直接クロックを供給します。外部接続は必須ではありませんが、テストやデバッグに役立ちます。</li> </ul>
INPUTXBAR または CLB_INPUTXBAR × 1	モジュール、I/O	SPI PICO ピンを CLB 入力に接続します。
OUTPUTXBAR または CLB_OUTPUTXBAR またはペリフェラル出力の CLB オーバーライド × 2	モジュール、I/O	<ul style="list-style-type: none"> <li>CLB 出力を ENCODER_CLOCK (MA) GPIO に接続します</li> <li>CLB を TxEN GPIO に接続します</li> </ul>
SPI × 1	モジュールと I/O	RS-485 の物理層データ信号を受信する 1 つの SPI インスタンス。SPI クロックは CLB によって制御されます。
CPU とメモリ	モジュール	CPU とメモリは各種の機能に使用されます。

#### 2.3.4.1 入力、出力信号、CLB タイル

このセクションでは、P65x デバイスで使用される入出力および CLB タイル接続について説明します。別のデバイスに移植する場合は、異なる配線または信号の使用が必要になることがあります。

- 実際に使用する GPIO ピンと SPI モジュールは、デバイス固有の LaunchPad のピン配置によって異なります。
- CLB との間の接続は、デバイスおよびピンの機能によって異なります。たとえば、CLB\_INPUTXBAR の代わりにデバイス INPUTXBAR を使用するような場合です。
- 特定の CLB タイルのインスタンスは、タイルが SPICLK など他の信号をオーバーライドできるかどうかによって異なります。たとえば、SPI-D を使用する場合、このデザインは、SPI-D に直接アクセスできるタイルに実装するのが最適です。

#### 注

入力 / 出力の回路図において、

- 色付きの丸で囲まれた文字は、次のセクションの CLB タイルの図でページ外への接続を示します。
- 文字の後に **\_RE** が続く場合は、立ち上がりエッジを示します。次に例を示します。D は、エンコーダ クロックです。D\_RE は、同じ信号に、CLB の立ち上がりエッジフィルタが適用された後の信号です。
- G は、有限ステートマシンからの出力であり、FRAME\_STATE と呼ばれます。G は、G.s0 および G.s1 という 2 つの状態信号で構成されます。



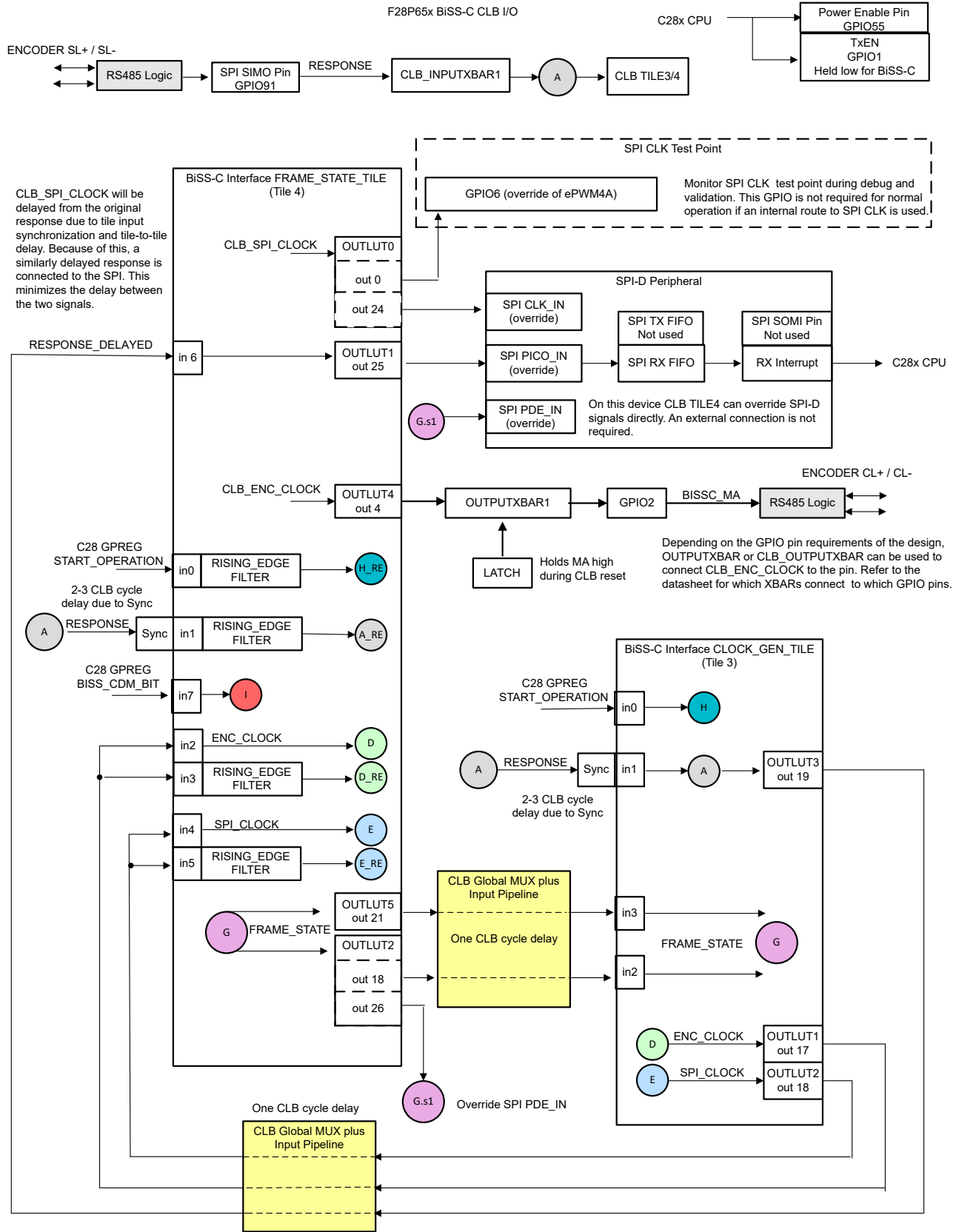


図 2-10. BOOSTXL\_POSMGR での F28P65 の 入力、出力、CLB の使用法

### 2.3.5 CLB BiSS-C 実装の詳細

CLB は、次の処理を行います。

- CDM ビットを使用してエンコーダ クロック MA 信号を生成する。
- SPI PICO 信号によりエンコーダからの応答を監視する。
- SPICLK を受信応答に同期する。
- SPI にクロックを供給して応答を受信する。

このセクションでは、次の 2 つの手法による CLB タイルの設計について説明します。

- 波形を使用して、トランザクションの各フェーズにおける CLB の動作を視覚化する。
- サブモジュールの相互接続を含む CLB タイルの設計。

#### 2.3.5.1 トランザクションの波形

CLB 設計を実装するときは、最初に波形を使用して必要な CLB の動作を視覚化すると便利です。これを行うには、まずトランザクションの例を考えます。BiSS-C のトランザクションは、MA 信号とエンコーダの応答で構成されることを思い出してください。送信は、[図 2-11](#) に示すように、いくつかの FRAME\_STATE に分割できます。最初の手順は、トランザクションの各要素を CLB サブモジュールにマッピングすることです。[表 2-4](#) にマッピングの例を示します。

表 2-4. BiSS-C のトランザクションから CLB へのマッピング

トランザクションの動作	CLB へのマッピング
FRAME_STATE の追跡	有限ステート マシン (FSM): 以前の状態と現在の入力を与えられたとき、新しい状態に遷移します。
所定の幅の 2 つのクロック信号 (MA 用と SPI クロック用) を生成。	各信号に対して、この要件はカウンタにマップされます。一致値を利用して、立ち上がり / 立ち下がりエッジのタイミングを生成します。その後、LUT (ルックアップ テーブル) が、このタイミングに基づいて実際のエッジを生成します。
MA および SPI クロックに対して所定の数のクロック パルスを送信します。	各クロックに対して、これはカウンタにマップされます。クロックのエッジでカウンタがインクリメントし、一致値がその数値に達したことを示します。
SPI クロックをエンコーダの応答に整列	SPI クロックを生成するカウンタは、エッジ遷移がエンコーダの応答と適切に整列するように構成できます。
エンコーダの応答を受信するときのみ SPI へのクロック供給を許可	LUT は、クロックが不要なときにクロックをブロックします。
BiSS タイムアウトの間、MA 信号を High または Low に保持 (CDM ビット)	GPREG 入力を使用して、CDM が High または Low のどちらであるべきかを、C28 から CLB に通知できます。これを LUT と組み合わせると、所定の数のクロックが送信された後、MA に値が保持されます。
TxEN の制御	BiSS-C の場合、TxEN は Low に保持されます。CLB 出力 LUT は、定数「0」を出力に適用できます。
MA を送信して、トランザクションを開始するように CLB に指示	C28x は、トランザクションの COUNTER と SPI を構成します。CLB GPREG により、C28x CPU は CLB 入力を直接変更してトランザクションを開始できます。

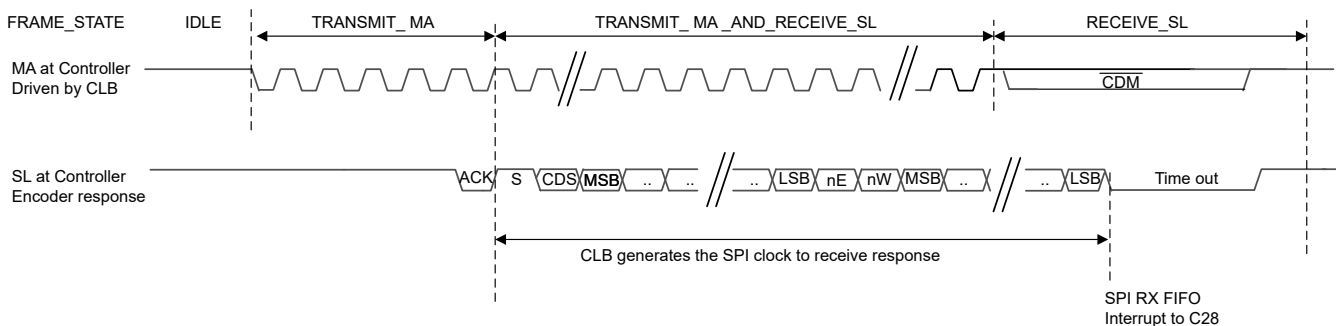


図 2-11. BiSS-C シングル サイクル データトランザクションの例

次の手順は、特定のサブモジュールの動作を視覚化することです。最初は簡単なスケッチを作成します。その後設計の進展に合わせて詳細を追加します。[図 2-12](#) は、CLB SystemC シミュレーション モデルを使用し、カスタム入力をエンコ

一ダの応答として生成された波形の例です。図 2-12 は実際の設計によって生成されましたが、この図の詳細は予備段階のスケッチと同様になっています。

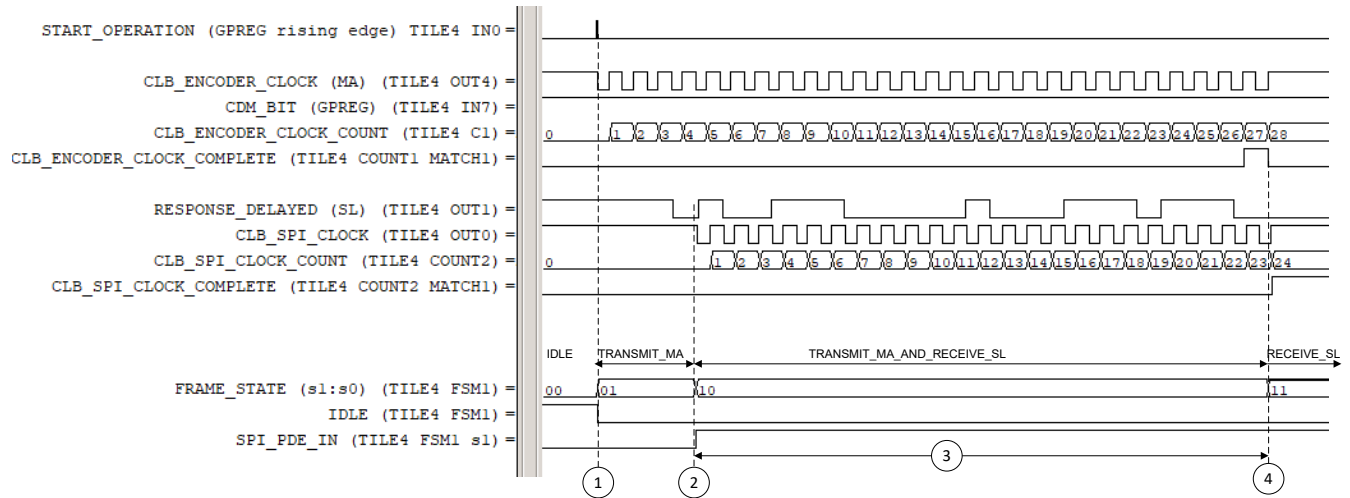


図 2-12. CLB 通信の波形

以下のセクションでは、図 2-12 のマーカー 1~4 を使用して、そのマーカーにおける設計の具体的な動作を説明します。マーカーには次のものがあります。

1. IDLE から TRANSMIT\_MA への遷移
2. TRANSMIT\_MA から TRANSMIT\_MA\_AND\_RECEIVE\_SL への遷移
3. TRANSMIT\_MA\_AND\_RECEIVE\_SL の期間中
4. RECEIVE\_SL への遷移

IDLE 時には、インターフェイス上でアクティビティはありません。C28x は、次の方法でトランザクションを開始する必要があります。

- CLB および SPI を構成する
- CLB タイルの GPREG レジスタにより START\_OPERATION 信号を High にプルする。GPREG は CLB の汎用レジスタで、C28x はこのレジスタを使用してタイルの入力を直接制御できます。

以下を参照してください。図 2-12、マーカー (1)

CLB の立ち上がりエッジフィルタがその入力に対してイネーブルなので、START\_OPERATION は 1 CLB\_CLOCK の間 High に維持されます。マーカー (1) において、メインステートマシン (Tile4 の FSM1) は、FRAME\_STATE を IDLE から TRANSMIT\_MA 状態に遷移させて応答します。

TRANSMIT\_MA の期間中:エンコーダ インターフェイスは MA 信号をエンコーダに送信します。この信号は、所定のデューティサイクルおよび定義されたクロック サイクル数を持つクロックです。サイクル数は、エンコーダの分解能によって異なります。MA にクロックを供給しているとき、CLB は SL ラインを監視してエンコーダの応答を検出します。

以下を参照してください。図 2-12、マーカー (2)

エンコーダの応答を検出するため、CLB は SL の立ち上がりエッジを監視します。立ち上がりエッジは、ACK 後のスタートビットに相当します。応答の検出に必要な時間は、一定のクロック数ではなく変動する可能性があり、エンコーダの状態やケーブル長に依存します。さらに、エンコーダが応答するのに余分な時間を必要とする場合、ACK 状態が延長されます。

エンコーダのスタートビットが検出されると、FRAME\_STATE は MA の送信と SL の受信に遷移します。

以下を参照してください。図 2-12、マーカー (3)

CLB は、SPI ペリフェラルへのクロック供給を開始して、応答を受信します。

- CLB は CLB\_SPI\_CLOCK を応答に整列させます。応答は、CLB が生成したクロックの立ち上がりエッジでサンプリングされます。
- CLB\_SPI\_CLOCK は、ペリフェラルへの入力を直接オーバーライドすることにより SPI\_CLK\_IN に接続されます。
- カウンタは、生成された SPI クロックの数を追跡します。このカウンタの match1 は、必要な SPI クロックが送信されたときを示します。SPI の FIFO が FIFO 割り込みレベルになるまでデータを満たすために、応答の長さに必要な値を上回る追加の SPI クロックが必要になることがあります。

以下を参照してください。図 2-12、マーカー (4)

### 2.3.5.2 FRAME\_STATE の生成

このセクションでは、FRAME\_STATE の生成について詳しく説明します。FRAME\_STATE は、トランザクション内のいずれか特定の時点で、どのクロックが生成されるかを決定します。

FRAME\_STATE (FSM\_1 s1, s0) は、次の 4 つの状態を遷移します。

- IDEL:
  - アクティビティなし
  - CLB は START\_OPERATION 信号の立ち上がりエッジを待機して、エンコーダ クロックの送信を開始します。
- TRANSMIT\_MA:
  - エンコーダ クロックを送信します
  - エンコーダからの ACK の応答を監視します
- TRANSMIT\_MA\_AND\_RECEIVE\_SL
  - エンコーダ クロックの送信を続けます
  - SPI クロックを開始して、応答を受信します
  - この期間中は、エンコーダ クロックと SPI クロックの両方がアクティブになります。
- RECEIVE\_SL:
  - エンコーダ クロックの送信が完了しました
  - MA 信号において CDM ビットはアクティブです
  - SPI クロックを供給して、応答の受信を終了します。
  - 使用するデータの長さや SPI 幅に応じて、追加の SPI クロックを生成して、SPI 割り込みをトリガできます。

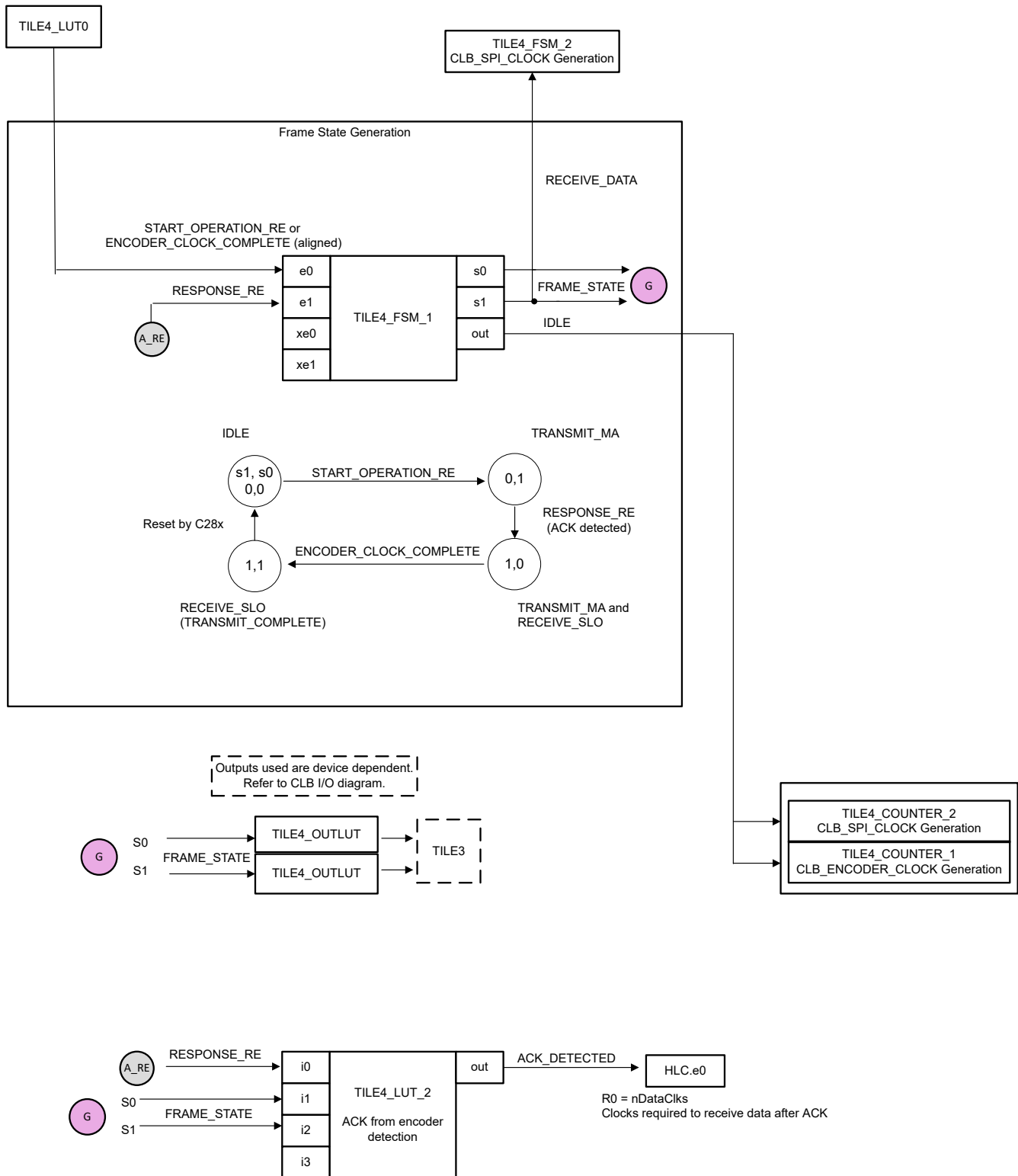


図 2-13. FRAME\_STATE の生成

対応する式を導出する方法の 1 つは、カルノー マップ (表 2-5 および 表 2-6) を使用することです。得られた式は OR 演算子で結合され、CLB ツールに入力されます。式を最も簡単な形に簡略化する必要はありません。



表 2-5. FRAME\_STATE FSM\_1 カルノー マップ、状態 s0

		現在の入力 (e1, e0) RESPONSE_RE、START_OPERATION、または ENCODER_CLOCK_COMPLETE			
		0.0	0.1	1.1 (1)	1.0
		以前の統計 s1, s0	0,0 IDLE	0	1 (2)
0,1 TRANSMIT_MA	1 (3)		1 (3)	0	0
1,1 RECEIVE_SL	1 (4)		1 (4)	1 (4)	1 (4)
1,0 TRANSMIT_MA RECEIVE_SL	0		1 (5)	1 (5)	0

- (1) 無効な、または予期しない e1:e0 の組み合わせに対応します。このような場合、システム設計者がステート マシンの動作を決定します。  
(2) !s1 & !s0 & e0  
(3) !s1 & s0 & !e1  
(4) s1 & s0. C28x は状態 1,1 から IDLE へ強制的に遷移します。  
(5) s1 & !s0 & e0

表 2-6. FRAME\_STATE FSM\_1 カルノー マップ、状態 s1

		現在の入力 (e1, e0) RESPONSE_RE、START_OPERATION、または ENCODER_CLOCK_COMPLETE			
		0.0	0.1	1.1	1.0
		以前の統計 s1, s0	0,0 IDLE	0	0
0,1 TRANSMIT_MA	0		0	1 (1)	1 (1)
1,1 RECEIVE_SL	1 (2)		1 (2)	1 (2)	1 (2)
1,0 TRANSMIT_MA RECEIVE_SL	1 (3)		1 (3)	1 (3)	1 (3)

- (1) !s1 & s0 & e1  
(2) s1 & s0. C28x は状態 1,1 から IDLE へ強制的に遷移します。  
(3) s1 & !s0

FSM\_1 からの OUT 信号は、IDLE 状態に対応します。

エンコーダの応答を検出することも、設計のもう 1 つの重要な要素です。図 2-13 に示す LUT\_2 は、エンコーダの ACK を検出します。FRAME\_STATE が WAIT\_FOR\_ACK (0,1) であり、かつ ENCODER\_RESPONSE\_RE が High になると、ACK が検出されたこととなります。これにより、次の式が得られます。i0 & (i1 & !i2):

- i0 == 1: ENCODER\_RESPONSE\_RE が high になる
- (!i2 & i1) == FRAME\_STATE == WAIT\_FOR\_ACK (0,1)

### 2.3.5.3 CLB\_SPI\_CLOCK の生成

CLB は、エンコーダの応答を検出し、SPI にクロックを供給してデータを受信する役割を果たします。最初にクロックのデューティと周波数を考慮します。図 2-14 に、SPI クロックを生成する CLB ロジックを示します。対応するシミュレーション波形を 図 2-15 に示します。

最初のステップは、指定された周波数とデューティを持つクロック (SPI\_CLOCK) を生成することです。SPI\_CLOCK の生成において:

- **TILE3\_COUNTER\_1**:クロックのデューティおよび周波数 (幅) を決めます。SPI\_CLOCK の High 遷移から Low 遷移までの時間は、CLB クロックの数で測定されます。以下の一致信号を使用します。
  - **zero match**:クロックの Low 遷移に対応
  - **match1**:クロックの High 遷移に対応
  - **match2**:クロックの幅に対応
- **TILE3\_FSM\_1:COUNTER\_1** の一致値 (**match1** および **zero match**) に基づいてクロック エッジを生成します
- **TILE3\_LUT\_0:COUNTER\_1 match2** 出力で示されるクロック幅に達したときにカウンタをリセットします

2 番目のステップは、クロックを受信データと整列させることです。これは、SPI がエンコーダの応答を正しく受信するために必要です。クロックの整列には、次の 2 つの課題があります。

1. [セクション 2.3.1.1](#) で説明しているように、ケーブル伝搬遅延のせいで、応答は任意の時刻に到着する可能性があります。
2. [セクション 2.3.1.2](#) で説明しているように、エンコーダは、アクリッジ (ACK) を遅延させて、追加の処理時間を要求することもできます。

クロックを整列させるために、LUT\_0 は、応答信号の立ち上がりエッジを監視します。最初の立ち上がりエッジは、アクリッジ (ACK) ビットの後のスタートビットに対応します。このエッジが検出されると、LUT\_0 は COUNTER\_1 をリセットし、SPI\_CLOCK を応答に整列させます。

3 番目のステップは、内部 SPI\_CLOCK に基づいて CLB\_SPI\_CLOCK を生成することです。図 2-15 に例を示します。

- SPI\_CLOCK は、SPI が応答を受信するのに必要とされるよりも長い時間生成されます
- CLB\_SPI\_CLOCK は、応答を受信するために必要なクロック数のみを出力します

これは、TILE4\_FSM\_2 を使って SPI\_CLOCK\_OUTPUT\_ENABLE 信号を生成することにより実現されます。この信号は、OUTLUT によって、必要に応じて SPI\_CLOCK を許可 / ブロックするために使用されます。

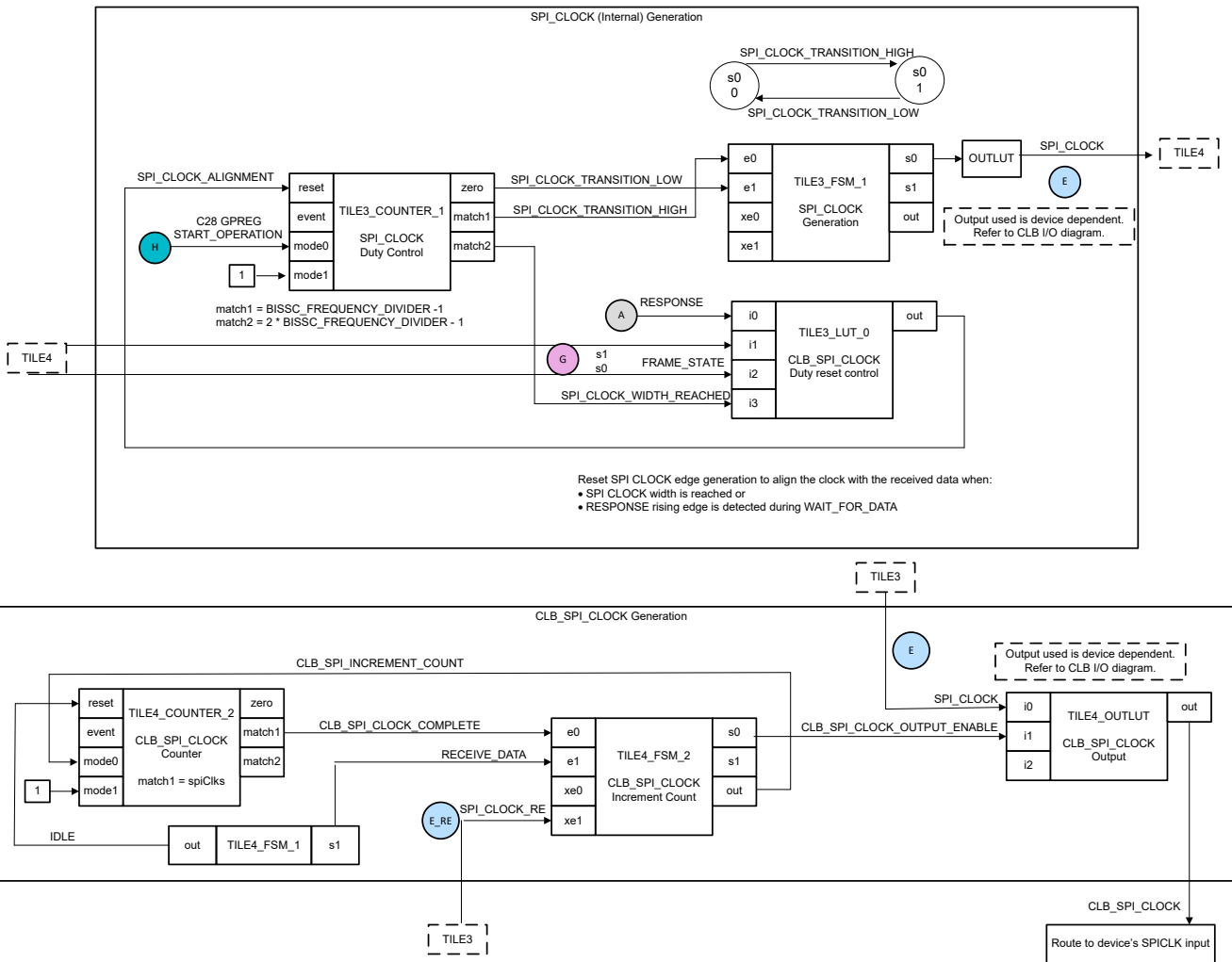


図 2-14. CLB\_SPI\_CLOCK の生成

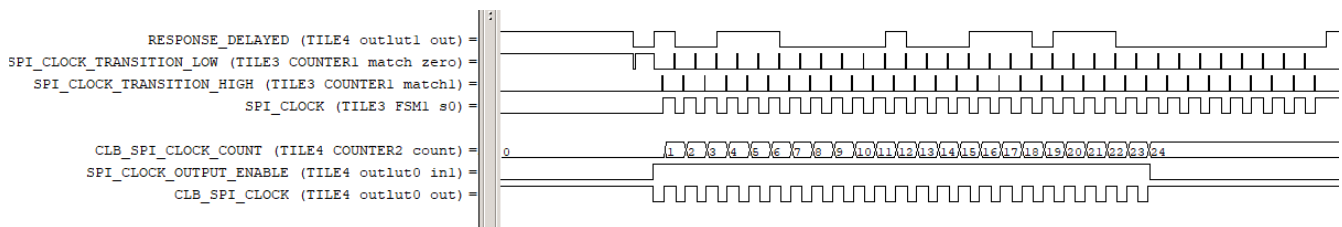


図 2-15. SPI クロック生成のシミュレーション波形

### 2.3.5.4 ENCODER\_CLOCK (MA) 生成

ENCODER\_CLOCK すなわち BISS MA の生成は、前のセクションで示した CLB\_SPI\_CLOCK の生成と似ています。2 つの主な相違点があります。

- 生成される MA クロックの合計数は、応答を受信するタイミングによって異なります。
- CDM ビットを示すために、MA 信号の最後は High または Low に保持される必要があります。

図 2-18 のマーカー (1) は、図 2-17 に示すロジックによってエンコーダの ACK が検出された時点を示します。マーカー (1) より前には、不明な数の MA クロックが生成されています。マーカー (1) 以降、必要な追加 MA クロックの数は  $X + 4$  です。ここで、

- X ビット: 位置ビットの合計 (シングルターン + マルチターン)
- 4 ビット: スタートビット + CDM ビット + エラービット + 警告ビット

• 6ビット:CRCビット

この数値は、アプリケーションによって TILE4 HLC レジスタ R0 に保存されます。HLC は、この値と現在のカウンタ値を使用して、クロックの総数 (COUNTER\_1 match1) を調整します。その例については、[図 2-18](#) を参照してください。

1. X = 13 であるとすれば、アプリケーションは 23 を HLC R0 に保存します
2. マーカー (1) で ACK が検出されると、HLC プログラムがトリガされます。
3. HLC が現在の CLB\_ENCODER\_CLOCK\_COUNT (4) を読み出し、R0 (23) を加算して 27 になります
4. HLC は、27 を TILE4 COUNT\_1 match1 にロードします
5. CLB\_ENCODER\_CLOCK\_COUNT が match1 に達すると、MA クロックの生成が完了します。

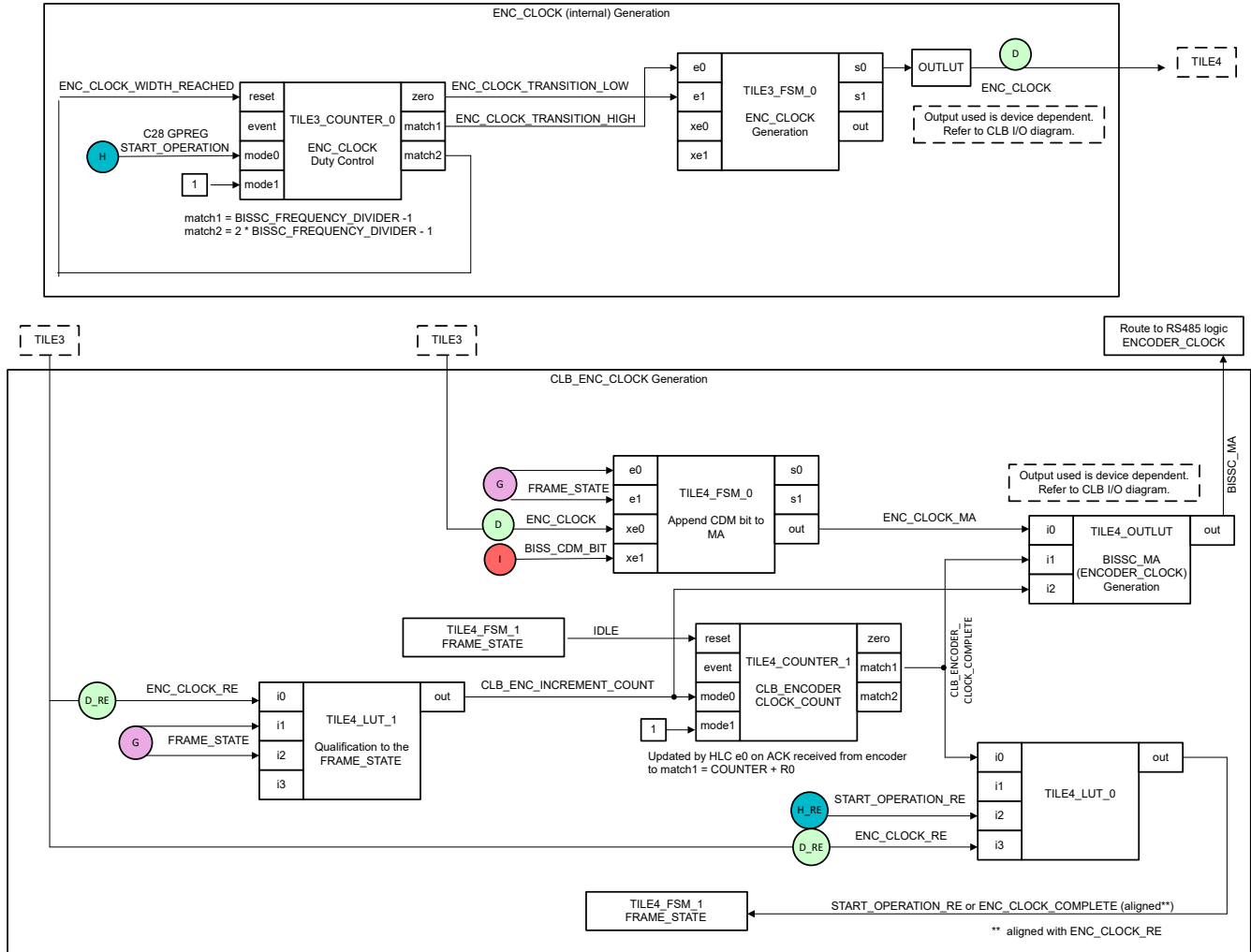


図 2-16. ENCODER\_CLOCK (MA) 生成

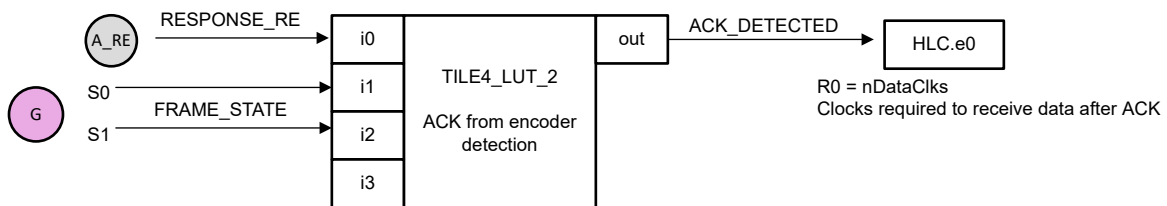


図 2-17. エンコーダの ACK 検出ロジック

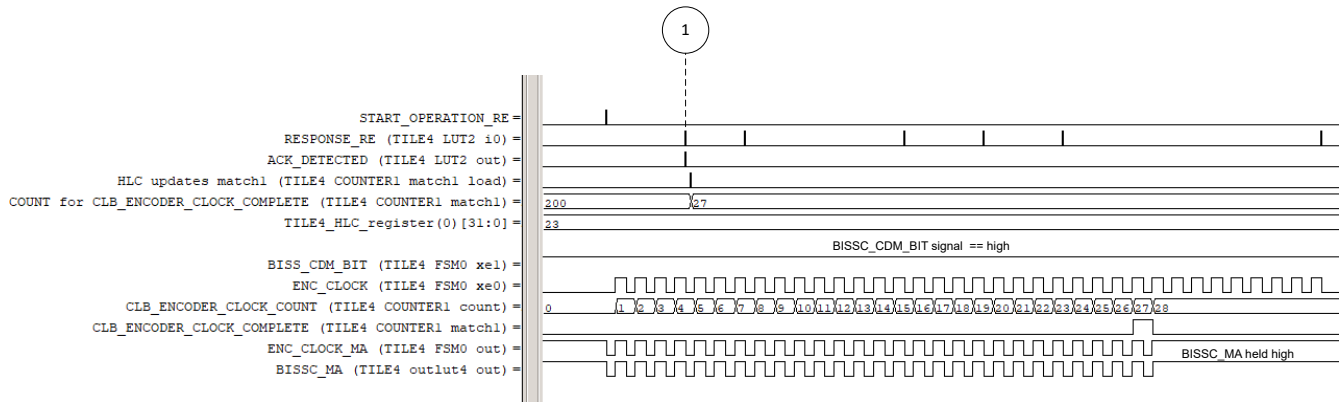


図 2-18. MA 生成シミュレーション波形:CDM 信号 High

CDM ビットは、エンコーダの各 BiSS-C フレームに転送される 1 ビットの制御データです。現在の CDM ビットを示すために、MA 信号の最後は High または Low に保持される必要があります。図 2-18 に、BISS\_CDM\_BIT のレベルが High である場合のシミュレーション波形を示します。図 2-19 に、BISS\_CDM\_BIT 信号が Low である場合のシミュレーション波形を示します。BISS\_CDM\_BIT 信号は、タイトルの GPREG への C2x CPU 書き込みによって制御されます。

注

この説明において、信号の状態 (High または Low) というのは、BISS\_CDM\_BIT 信号のレベルを示します。実際の CDM ビット値は、エンコーダによって信号レベルの反転として解釈されます。たとえば、BISS\_CDM\_BIT 信号が High の場合、エンコーダは CDM ビット値を 0 として検出します。ソフトウェア ライブラリはこの反転を考慮しています。

Tile 4 FSM\_0 は、CLB\_ENCODER\_CLOCK\_COMPLETE により FRAME\_STATE が変化したときに、MA 信号に BISS\_CDM\_BIT レベルを付加します (図 2-16)。BISS\_CDM\_BIT のレベルが Low の場合は、出力で ENC\_CLOCK\_MA から余分なエッジが除去されます (図 2-19)。

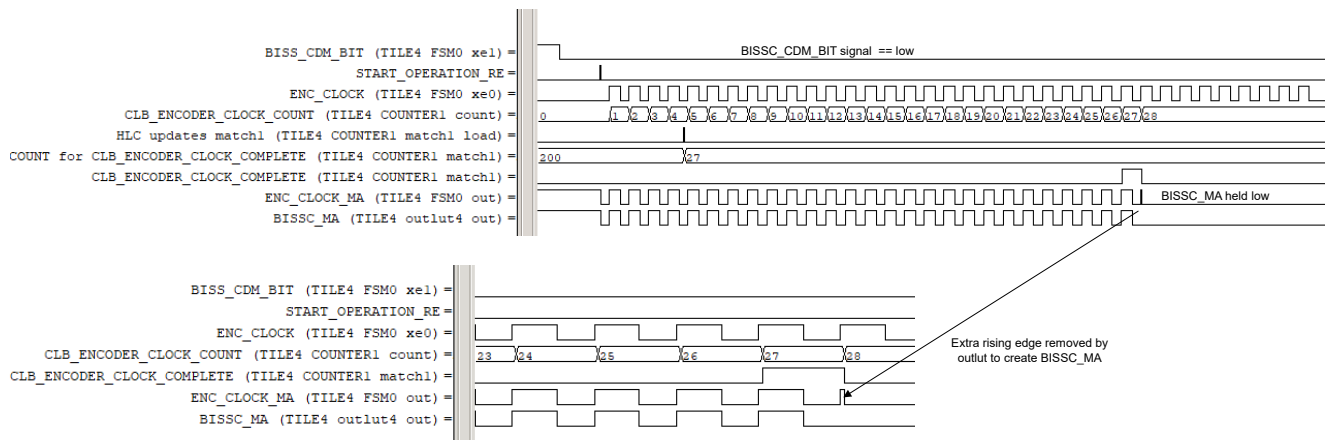


図 2-19. MA 生成シミュレーション波形:CDM 信号 Low

### 2.3.6 PM BiSS-C インターフェイス ライブラリ

PM BiSS-C エンコーダ インターフェイス ライブラリには、次のものが含まれています。

- CLB ロジックの実装。
- 適切に定義されたアプリケーション プログラミング インターフェイス (API)。これにより、C2000 デバイスが BiSS-C ポジション エンコーダと通信可能。

このセクションでは、API が提供する関数の概要について説明します。



### 2.3.6.1 PM BiSS-C ライブラリ関数

BiSS-C ライブラリは以下の機能で構成されており、これによりユーザーは BiSS-C エンコーダとのインターフェイスを確立できます。API の詳細な説明については、『C2000 BiSS-C エンコーダ インターフェイスのソフトウェア ガイド』([html](#)、[pdf](#)) を参照してください。表 2-7 に、API の概要を示します。

**表 2-7. BiSS-C ライブラリ関数**

名称	説明
<b>初期化関数</b>	
CRC テーブルの生成	指定されたビット数を持つ指定された CRC 多項式に対して、ルックアップ テーブルを生成します。2 つのテーブルが使用されます。1 つはシングル サイクル データ CRC 計算用、もう 1 つはコマンド データ CRC 計算用です。
ペリフェラルの設定	CLB ロジックによって実現される BiSS-C サブシステムを構成します。
パラメータ初期化	シングル サイクル データから位置および CRC を抽出するために必要なパラメータを初期化します。
<b>ランタイム関数</b>	
動作の開始	MA 信号を開始して BiSS-C 転送を起動します。
コマンド データの処理	それぞれの BiSS-C フレームで CDM 出力と CDS 入力を処理するコマンド データ ステートマシン。
位置の受信	SCD から位置、エラー、警告、CRC を抽出します。CRC を計算し、受信した CRC と比較します。
SCD トランザクションの設定	新しいシングル サイクル データ転送のために、SPI と CLB をセットアップします。

### 3 ハードウェア、ソフトウェア、テスト要件、テスト結果

このセクションでは、テスト手順とテスト結果について説明します。以下のものが含まれます。

- ハードウェア要件
- ソフトウェア要件
- テスト結果
- ベンチマーク
- [トラブルシューティング ガイド](#)

#### 3.1 ハードウェア

TIDM-1010 で実験を行うには、次のハードウェア コンポーネントが必要です。

- TIDM-1010 BoosterPack (BOOSTXL-POSMGR とも呼ばれます)
- 外部の 5V DC 電源 ([表 1-1](#) を参照)
- F28P65x LaunchPad 開発キット (LAUNCHXL-F28P65X) および USB ケーブル
- BiSS-C エンコーダ
- エンコーダと TIDM-1010 を接続するためのケーブル
- メス終端ケーブルをワイヤリード アダプタに接続するためのカスタム アダプタ
- CCS がインストールされているコンピュータ

### 3.1.1 TIDM-1010 のジャンパ構成

TIDM-1010/BOOSTXL-POSMGR 基板のジャンパ構成を、[図 3-1](#) に示します。

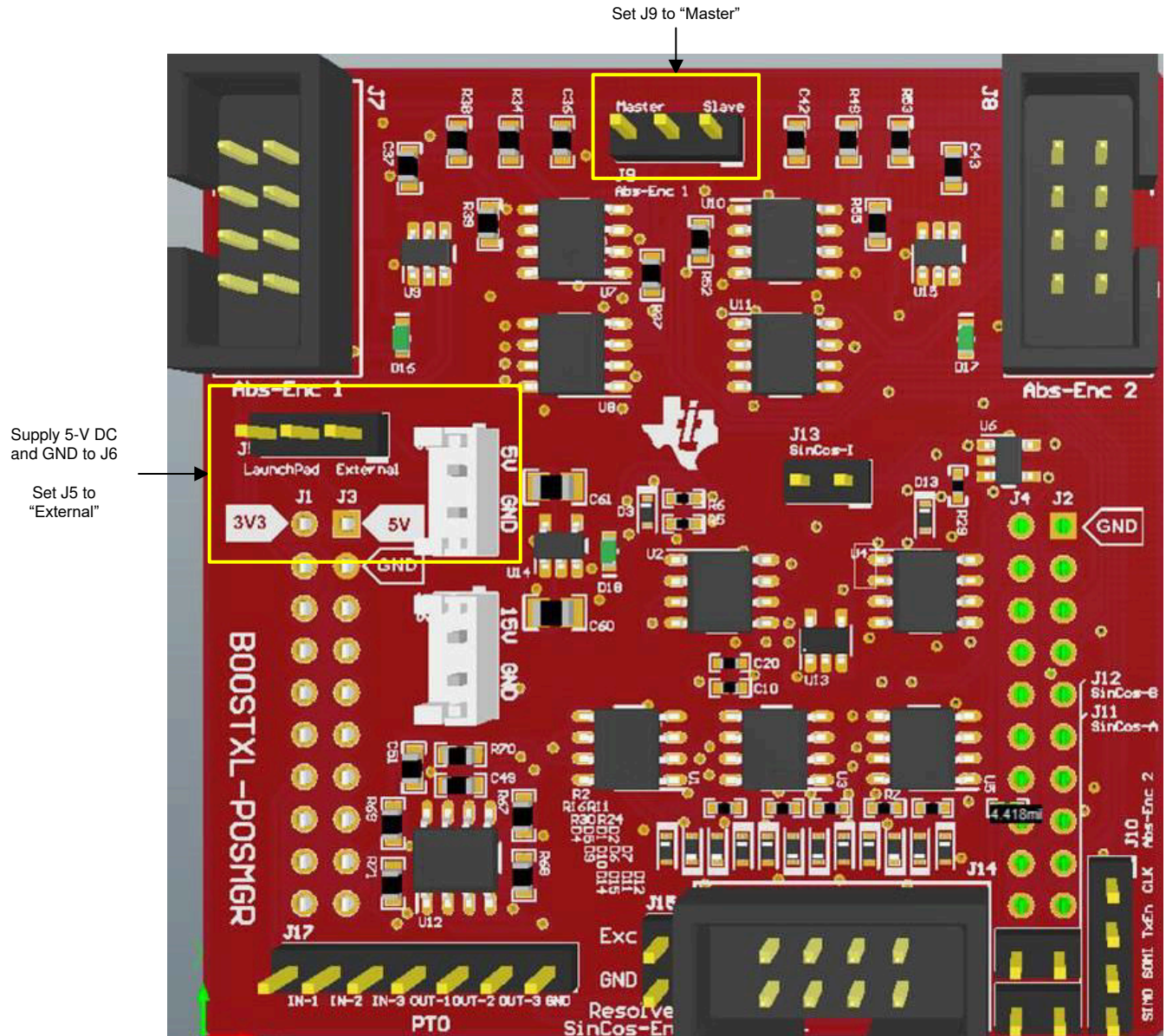


図 3-1. TIDM-1010 基板のジャンパ構成

TIDM-1010 基板のジャンパ構成を、表 3-1 に示します。

**表 3-1. TIDM-1010 基板のジャンパの詳細**

ジャンパ	機能	位置
J5	TIDM-1010、5V、電源プレーンソースの選択	外部 (1)
J9	Abs-Enc-1 マスタ / スレーブ モードの選択	マスタ (2)
J11	サイン - コサイン エンコーダ - A 信号イネーブル	オープン
J12	サイン - コサイン エンコーダ - B 信号イネーブル	オープン
J13	サイン - コサイン エンコーダ - インデックス信号イネーブル	オープン

- (1) この構成では、図 3-1 に示すように、ユーザーは J6 に外部電源を供給する必要があります。  
 (2) スレーブ モード オプションは使用しないでください。このモードの **BoosterPack** ロジックにはエラーがあります。

### 3.2 ソフトウェア

このセクションでは、TIDM-1010 で使用されるソフトウェアの概要について説明します。包括的なドキュメントについては、『C2000 BiSS-C エンコーダ インターフェイスのソフトウェア ガイド』(HTML、PDF) を参照してください。

このソフトウェア ガイドには、次の資料が含まれています。

- システム デモ コードの資料
- BiSS-C アプリケーション プログラマ インターフェイス (API) の資料
- 独自のソリューションにライブラリを組み込む
- デュアル CPU デバイス上で C28x CPU1 から CPU2 にソリューションを移植するためのガイド。
- ソフトウェア変更履歴

このリファレンス デザインに実装されているソフトウェア アーキテクチャを、図 3-2 に示します。このソフトウェアは、モジュール形式やポータブルな形式で実装されています。主なコンポーネントには、C2000 ドライバ ライブラリ、BiSS-C エンコーダ インターフェイス ライブラリ、SysConfig GUI デバイス構成ツール、CLB ツールが含まれています。

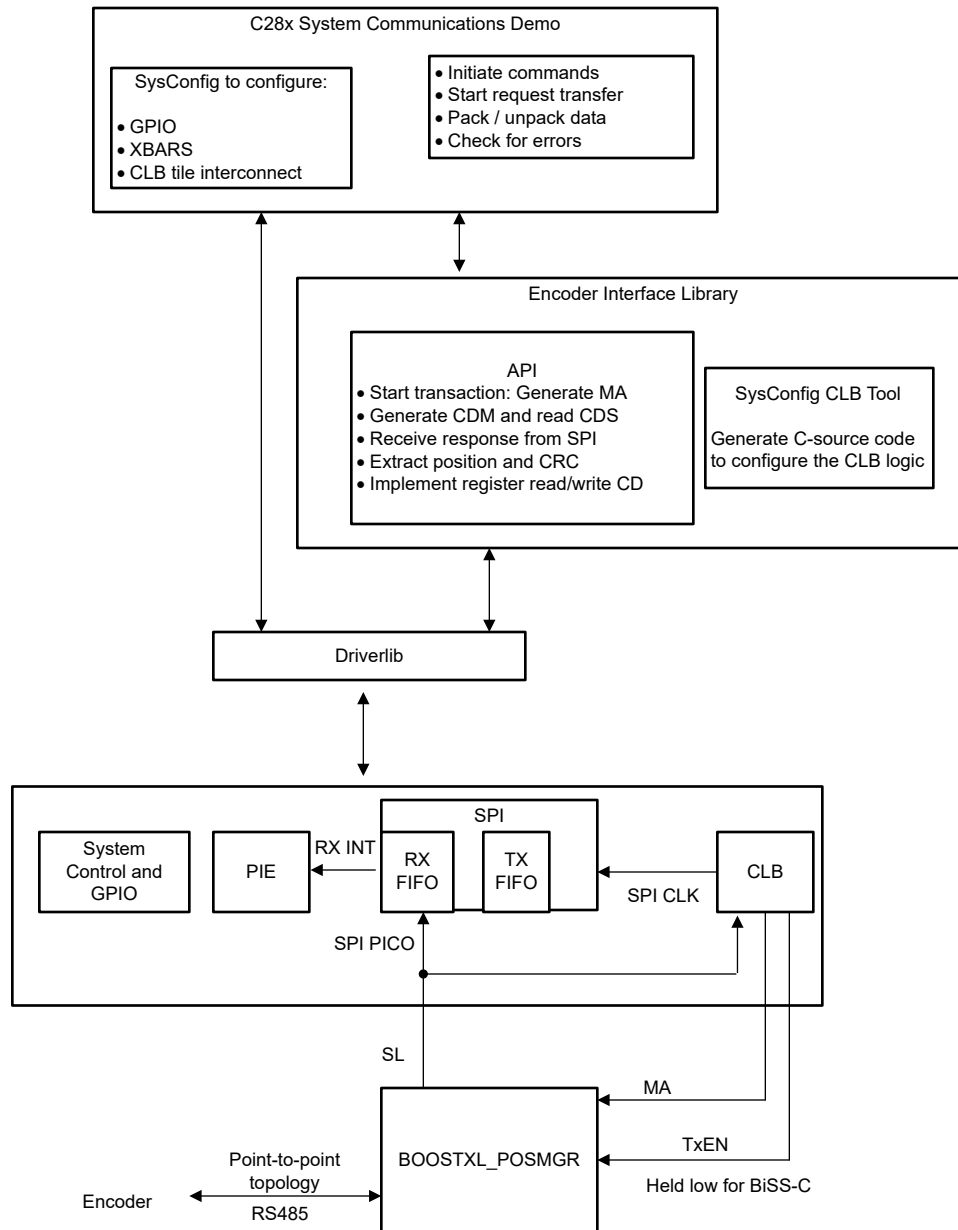


図 3-2. ソフトウェア アーキテクチャ

C2000 BiSS-C 通信デモのフローチャートを、[図 3-3](#) に示します。サンプル アプリケーションは、C2000 デバイスの構成、制御データ フレームの生成、MA 信号の送信、応答の展開、CRC のチェックを行います。



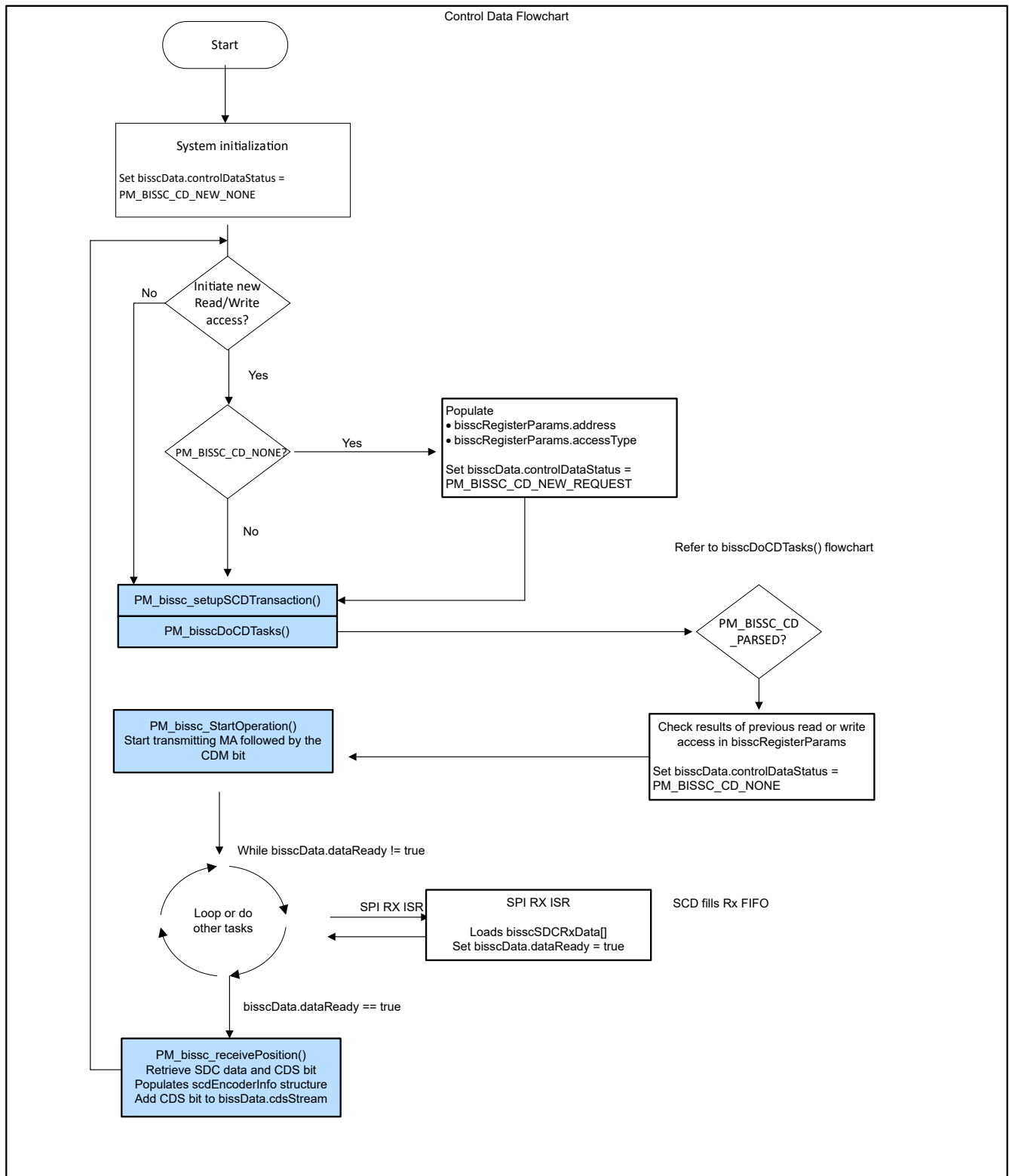


図 3-3. 通信デモのフローチャート

### 3.2.1 C2000 ドライバライブラリ (DriverLib)

C2000 ドライバライブラリ (Driverlib) は、C2000 デバイス ファミリー用の一連の低レベル API です。Driverlib には、メモリ マップされたペリフェラル レジスタを構成するための、使いやすい関数呼び出しが用意されています。Driverlib の完全な

ソースは、C2000Ware および [C2000Ware モーター制御 SDK](#) に含まれています。詳細については、『[C2000 ソフトウェアガイド](#)』の「DriverLib」のセクションを参照してください。

### 3.2.2 C2000 SysConfig

C2000 SysConfig は、C2000 リアルタイム制御 MCU を構成するためのグラフィカル ユーザー インターフェイス ツールです。SysConfig は、DriverLib に接続する組込みソフトウェアを自動生成します。このリファレンス デザインは、SysConfig ツールを使用して、SPI、GPIO、INPUTXBAR/OUTPUTXBAR、CLB MUX を構成するコードを生成します。詳細については、『[C28x Academy: SysConfig モジュール](#)』をご覧ください。

### 3.2.3 C2000 構成可能ロジックブロック ツール

C2000 CLB ツールを使用すると、グラフィカル インターフェイスを使用して CLB ロジックを構成できます。CLB ツールは、Code Composer Studio に組み込まれている使いやすい GUI であり、C2000 SysConfig プラグインを使用します。このリファレンス デザインでは、設計の説明に記載されているように、CLB ツールを使用して、BiSS-C エンコーダ インターフェイスのタイルを構成します。詳細については、『[C28x Academy: 構成可能ロジックブロック モジュール](#)』を参照してください。

### 3.2.4 Code Composer Studio™ と C2000WARE-MOTORCONTROL-SDK のインストール

1. CCSTUDIO がコンピュータにインストールされていない場合は、[CCSTUDIO IDE v12.7.1](#) またはそれ以降をインストールします。
2. [C2000WARE-MOTORCONTROL-SDK v5.03.00.00](#) またはそれ以降がコンピュータにインストールされていない場合は、インストールします。
3. インストール後の手順については、『[C2000 BiSS-C エンコーダ インターフェイスのソフトウェア ガイド](#)』([html](#)、[pdf](#)) を参照します。

#### 注

CLB ツールは、Code Composer Studio (sysconfig) と、SDK の C2000Ware サブコンポーネント (サポートユーティリティ) に含まれています。CLB ベースのシミュレーションを実行するには、『[CLB ツール ユーザー ガイド](#)』に記載されている追加ツールをインストールする必要があります。

### 3.2.5 リファレンス ソフトウェアの場所

このリファレンス デザインに含まれるソフトウェアは、次の 2 つの部分に分けられます。

- エンコーダ インターフェイスの使用方法を示すサンプル システム。サンプル プロジェクトのソース ファイルの場所を、[表 3-2](#) に示します。
- エンコーダ インターフェイス ライブラリ。ライブラリのソース ファイルの場所を、[表 3-3](#) に示します。

包括的なドキュメントについては、『[C2000 BiSS-C エンコーダ インターフェイスのソフトウェア ガイド](#)』([HTML](#)、[PDF](#)) を参照してください。

**表 3-2. サンプル システムの場所**

C:\ti\c2000\c2000ware_MotorControl_SDK_[version]\	SDK のデフォルトのインストール場所。([SDK])
[SDK]\solutions\boostxl_posmgr\	デバイス固有の基本インストール ディレクトリ ([pm_base])
[pm_base]\[device]\ccs\bissc	Code Composer Studio (CCS) の projectspec ファイル。プロジェクトを CCS ワークスペースにインポートするために使用します。

**表 3-3. エンコーダ インターフェイス ライブラリの場所**

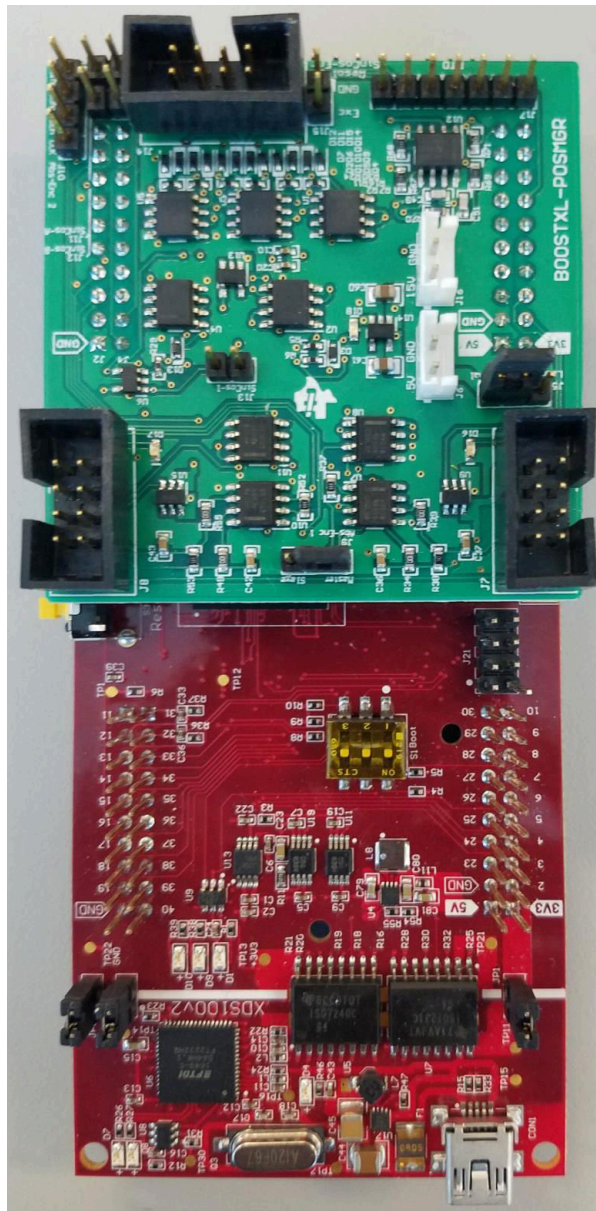
C:\ti\c2000\c2000ware_MotorControl_SDK_[version]	SDK のデフォルトのインストール場所。([SDK])
[SDK]\libraries\position_sensing\bissc	ライブラリの基本インストール ディレクトリ ([lib_base])
[lib_base]\ccs\[device]	リファレンス ライブラリ用の Code Composer の projectspec ファイル。これらのプロジェクトを使用して、各デバイスのライブラリを再ビルドします。

### 3.3 テストと結果

このセクションでは、テスト手順、結果、ベンチマークについて詳しく説明します。トラブルシューティングのガイドも紹介します。

#### 3.3.1 ハードウェアの構成

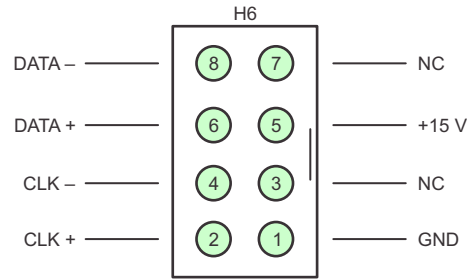
1. TIDM-1010 デバイスのジャンパ構成が [表 3-1](#) の記述と一致していることを確認する。
2. BoosterPack サイト 2 コネクタ (J1、J3 および J4、J2) を使用して、TIDM-1010 デバイスを LaunchPad に接続する。[図 3-4](#) に示すように、TIDM-1010 デバイスが LaunchPad のサイト 2 に接続されていることを確認する。



**図 3-4. LaunchPad™ のサイト 2 に接続された TIDM-1010 ボード**

3. USB ケーブルを LaunchPad に接続する。
4. エンコーダへの接続:
  - a. 丸形メス コネクタを使用してケーブルを BiSS-C インターフェイスに接続するアダプタを用意し、リード アダプタを配線する (エンコーダのコネクタ J7 に使用するヘッダーについては、BOM を参照)。
  - b. CLK+/CLK- をエンコーダの BiSS-C MA+/MA- 信号に接続する。
  - c. DATA+/DATA- をエンコーダの応答 SL+/SL- に接続する。

- d. 前の手順で作成したアダプタのヘッダーを挿入して、ABS-Enc-1 (J7) に接続する。ケーブルのメス側はエンコーダに接続する。J7 のピン配置を、[図 3-5](#) に示す。

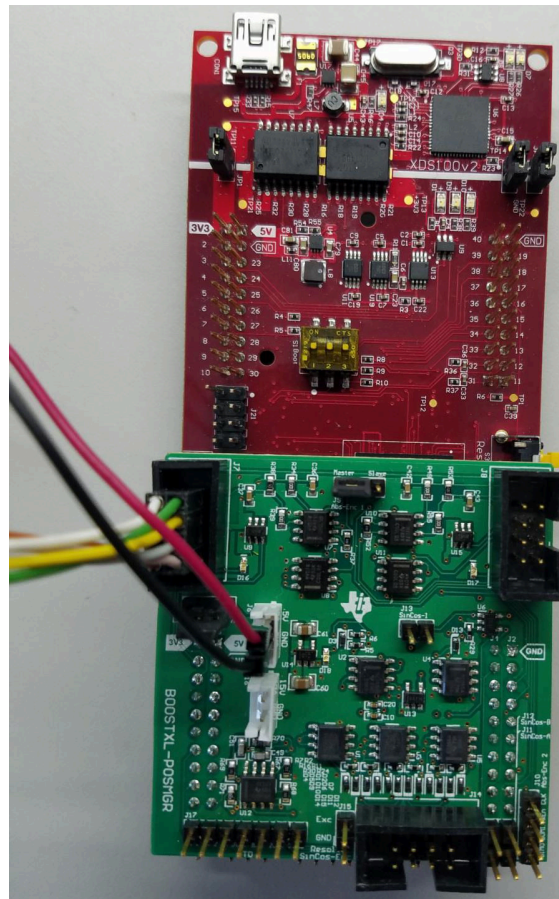


**図 3-5. TIDM-1010 基板の Abs-Enc-1 (J7) ピン配置**

5. [図 3-1](#) に示すように、J6 に 5V DC と GND を供給する。基板は [図 3-6](#) のようになる。BoosterPack LED D18 が点灯し、基板に電力が供給されたことが示される。

**注**

一部のエンコーダでは、電源投入時に **BoosterPack** から十分な電流が得られない場合があります。エンコーダが応答しない場合は、**BoosterPack** の外部電源をエンコーダに接続してみてください。これを行う場合、共通のグラウンドを **BoosterPack** に接続します。



**図 3-6. BiSS-C エンコーダに接続された TIDM-1010 基板**



### 3.3.2 プロジェクトのビルドおよびロード

『C2000 BiSS-C エンコーダ インターフェイスのソフトウェア ガイド』(HTML、PDF) の指示に従い、システム ソリューションをロードして実行します。ソフトウェア ガイドの「システム ソリューション」セクションを参照してください。

このセクションには、次の手順が記載されています。

- プロジェクトをデバイスの CCS (Code Composer Studio) にインポートする
- ライブラリとサンプル システムを構成する
- ビルド構成を選択する
- ウォッチ (数式) ウィンドウに値を入力する
- コードを実行する

### 3.3.3 実行するサンプルコード

BiSS-C システム ソリューションは、通信専用のデモです。このデモは、MA 信号をエンコーダへ送信し、応答を受信して、エラーをチェックします。このパターンが `while (1){}` ループで繰り返されます。さらに、このデモは、読み取り / 書き込みアクセスをエンコーダ レジスタへ定期的送信します。提供された状態では、バンク選択レジスタが使用されます。デモの実行中に、手動でモーターまたはエンコーダのシャフトを回転させながら、ロジック アナライザまたはスコープを使用して、MCU の出力信号を監視できます (図 3-7)。

#### 注

F2837xD についてのみ、CLB と SPI クロックの間に外部接続を必要とします。その他のデバイスには、CLB と SPICLK の間に内部接続があります。内部接続があるデバイスでは、SPICLK をピンから外に取り出して監視に使用することもできます。SPI CLK のテスト接続を、セクション 2.3.4.1 のデバイス入出力図に示します。

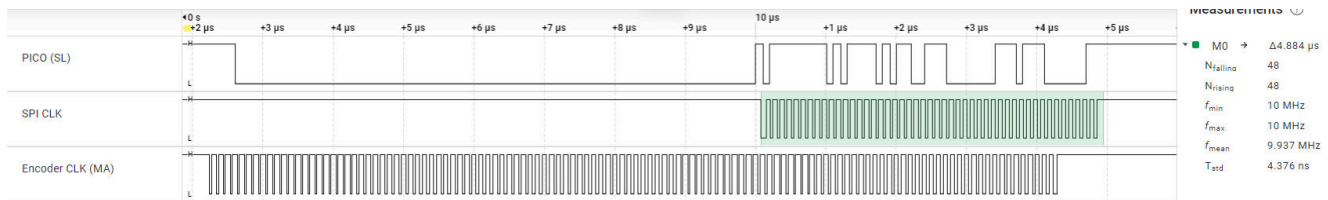


図 3-7. BiSS-C 波形

図 3-7 に示す波形は、次の条件によるものです。

- 32 ポジション ビットを持つエンコーダ
- 10MHz MA 信号で 10m ケーブル
- 監視のために SPI クロックをピンに出力
- SPI FIFO は、12 ビットワードで、レベル 4 がフルのときに割り込みが発生するように構成
- 割り込みをアサートするためには 48 クロックが必要
- サンプリング周波数 100MHz のロジック アナライザによってキャプチャ

### 3.3.4 エンコーダのテスト

表 3-4 に、各種エンコーダによるテストを示します。これらのテストは、テキサス・インスツルメンツで実施しました。テストには、位置の読み取り、レジスタの読み取り、レジスタへの書き込みが含まれます。最大 10m のケーブル長で 10MHz 動作が可能です。低い周波数で、より長いケーブル長もテストしました。

最大 MA 周波数は、エンコーダとエンコーダ ケーブルの両方に依存します。ケーブルの品質は、エンコーダのデジタル通信性能および通信可能距離に影響を与えます。たとえば、長いケーブルの電圧降下を最小限に抑えるために、エンコーダケーブルの電源線には太いゲージの電線を使用することがあります。通常、エンコーダ ベンダは、そのエンコーダに対応するケーブルを推奨しています。100m 以上のケーブル長で動作する場合には、このケーブルの品質が不可欠です。

MA クロック周波数とケーブル長との関係のガイドラインについては、ドキュメント『BiSS インターフェイス: AN15: BiSS C マスタ動作の詳細』を参照してください。これは、BiSS インターフェイスのウェブサイトから入手できます。

表 3-4. エンコーダ テストの概要

エンコーダのメーカー	エンコーダ名	タイプ	分解能 (ビット数)
Lika	HS58S18	ロータリー	18 ビット
Lika	EH036-20-12	ロータリー	32 ビット (12 MT + 20 ST)
iCHaus	EVAL MHM1D	評価基板	28 ビット (16 MT + 12 ST)

### 3.3.5 ベンチマーク

BiSS-C ライブラリ関数を RAM から実行するために必要な C28x CPU サイクル数を、表 3-5 に示します。このデータは、ERAD モジュールと次のコンパイラ設定を使用して収集されました。

- C2000 Codegen ツール V22.6.1.LTS
- -O2 -mf2
- float\_support: fpu32
- tm\_u\_support: tm\_u0
- fp\_mode: relaxed
- abi: eabi

#### 注

これらの関数は、アプリケーションによっては、さらに最適化できます。たとえば、エンコーダの分解能が実行時ではなくコンパイル時にわかっている場合、一部のコードを除去できます。

表 3-5. サイクル数ベンチマーク

機能	サイクル -O2 -mf2	注
PM_bissc_setupSCDTransaction	369	
bissc_setupCDTest	23	
PM_bissc_receivePosition	552	エンコーダの分解能に依存します。この測定には、12 のマルチターンと 20 のシングル ターン ビットを持つエンコーダを使用しました。
PM_bissc_doCDTasks	39 (最小)、225 (最大)	最大サイクルは、レジスタの読み取りまたはレジスタの書き込みが完了したときに発生します。
PM_bissc_startOperation	39	
PM_bissc_setFreq	73	
bissc_getCRC	90	
bissc_spiRxlSR	497	

各ライブラリ ソース ファイルに対応するコード サイズを、16 ビット ワード単位で表 3-6 に示します。C28x ルックアップ テーブルは 256 ワードの RAM またはフラッシュを必要としますが、この表には反映されていません。

表 3-6. 16 ビット ワード単位のコード サイズ

ソース ファイル	コード サイズ -O2 -mf2
pm_bissc_source	910
pm_bissc_crc	49
clb_config	463

### 3.3.6 トラブルシューティング

次の波形を調べると、トラブルシューティングに役立つことがあります。設計の説明に記載されている I/O 図を参照してください。

- CLB で生成される SPI クロック。これは、観察のためにテストピンに出力されます。
- SPI 入力ピンでのエンコーダからの応答。
- RS485 ラインドライバとエンコーダの間でのエンコーダからの応答 (SL+/SL-)。注: このデータは差動信号です。したがって、観察には特別なプローブが必要です。



- TxEN 信号。この信号が **Low** のままであることを確認してください。BiSS-C 設計では、TxEN を **High** にプルしませ  
ん。
  - RS485 ラインドライバとエンコーダの間のエンコーダ クロック (MA+/MA-)。注:このデータは差動信号です。したがっ  
て、観察には特別なプローブが必要です。
1. MA 信号が送信されない場合:
    - 問題が RS485 ラインドライバの前か後かを確認します。
    - ラインドライバとエンコーダの間で問題がある場合、TxEN が **Low** に保持されていることを確認します。
  2. MA 周波数が想定どおりでない場合:
    - `bissc.h` の `BISSC_MA_CLOCK` の定義および `BISSC_FREQ_DIVIDER` の定義を確認します
  3. エンコーダの応答が見られない場合:
    - エンコーダへの電源接続を確認します。
    - 電源電流がエンコーダの仕様に適合するかどうかを確認します。
    - MA の周波数を下げて、もう一度やり直してください。これは、ケーブルまたはコネクタの問題を示している可能性  
があります。
    - ケーブルの設計および長さがエンコーダの製造要件を満たしていることを確認します。
    - 問題が RS485 ラインドライバの前か後かを確認します。
  4. 応答波形が観測されたが SPI でキャプチャされていない場合:
    - これは、CLB の設計を別のタイルに移動した場合、または別のデバイスに移植した場合に発生する可能性があります  
ます。
      - CLB からの SPI クロックを確認します。このクロックは、CLB がエンコーダからの応答を検出すると開始されま  
す。内部クロスバー接続が、SPI および適切な CLB 入力の両方に応答を正しく配線していることを確認しま  
す。
      - SPI クロックが観測される場合、CLB 出力からの内部接続が、使用されている SPI インスタンスに行っているこ  
とを確認します。これは、使用されている CLB タイルによって異なります。

## 4 デザイン ファイル

設計ファイルをダウンロードするには、[TIDM-1010](#) のプロダクト ページを参照してください。

## 5 ソフトウェア ファイル

このソフトウェア ソース ファイルは、[C2000™ MCU](#) 向け [MotorControl](#) ソフトウェア開発キット (SDK) に付属しています。

## 6 関連資料

1. iCHaus. [iC-Haus](#)、ウェブサイト
2. BiSS インターフェース [BiSS インターフェース](#)、ウェブサイト。
3. 『C2000 BiSS-C エンコーダ インターフェイスのソフトウェア ガイド』(HTML、PDF)
4. ウィキペディア、[巡回冗長検査](#)、ページ。
5. テキサス・インスツルメンツ: [Code Composer Studio](#) ページ。
6. テキサス・インスツルメンツ: [C28x Academy](#) は、すべての C28x デバイスのさまざまなトピックを取り扱った使いやすいトレーニング モジュールを公開しています。
7. テキサス・インスツルメンツ: 『[C2000 ソフトウェア ガイド](#)』には、C2000 ソフトウェア、ソフトウェア開発キット、開発ツールの概要が記載されています。
8. テキサス・インスツルメンツ: [C2000 DesignDRIVE](#)、産業用ドライブおよびモーター制御向けソフトウェア
9. テキサス・インスツルメンツ: 『[C2000 Position Manager SinCos ライブラリ](#)』、ユーザー ガイド

## 商標

C2000™, BoosterPack™, LaunchPad™, and E2E™ are trademarks of Texas Instruments.

BiSS-C™ is a trademark of iC-Haus GmbH.

すべての商標は、それぞれの所有者に帰属します。

## 7 用語

<b>C28x</b>	C28x CPU コアを搭載したデバイスを指します。
<b>API</b>	アプリケーション プログラミング インターフェイス。エンコーダ インターフェイス プロトコルを実現するライブラリの定義。
<b>BiSS</b>	センサおよびアクチュエータ向けのオープンソース デジタル インターフェイス。このドキュメントにおいて、BiSS インターフェイス または BiSS は、2007 年に導入された BiSS-C プロトコルを意味します。BiSS-C は双方向シリアル同期連続モードの略称です。プロトコルと実装の詳細については、 <a href="#">BiSS インターフェイス</a> のウェブサイトを参照してください。
<b>BiSS フレーム</b>	シングル サイクルのデータおよび 1 ビットの制御データ。BiSS フレームは、BiSS サイクルごとに応答として送信されます。
<b>BiSS ライセンス</b>	テキサス・インスツルメンツ MCU のご購入には、BiSS IP を使用するためのライセンスが自動的に含まれていません。BiSS インターフェイスのユーザーは、 <a href="#">BiSS インターフェイス</a> から独自のライセンスを申請する必要があります。
<b>CD</b>	BiSS-C 仕様に基づくものです。制御データ。BiSS フレームごとに 1 ビット送信されるデータ。たとえば、CD は、エンコーダ レジスタの読み取りと書き込みを許可します。
<b>CDM</b>	BiSS-C 仕様に基づくものです。制御 データ マスタ。各 BiSS フレームで、1 ビットの制御データ CD がコントローラからエンコーダへ送られます。
<b>CDS</b>	BiSS-C 仕様に基づくものです。制御データ スレーブ。各 BiSS フレームで、1 ビットの制御データ CD がエンコーダからコントローラへ送り返されされます。
<b>CLB</b>	C2000 デバイス上の構成可能ロジック ブロック ペリフェラル。
<b>CRC</b>	巡回冗長検査。

<b>CTS</b>	BiSS-C 仕様に基づくものです。制御選択ビット。BiSS-C 制御データ (CD) フレームは、スタートビットから始まり、次に CTS ビットが続きます。CTS が 0 の場合、制御フレームはコマンド フレームです。CTS が 1 の場合、制御フレームはレジスタ アクセス フレームです。
<b>MA</b>	BiSS-C 仕様に基づくものです。マスタ クロック。このパルス列の出力は、コントローラからエンコーダ / センサに送信されます。この信号は、エンコーダ クロックと、マスタからの 1 ビットの制御データで構成されます。
<b>Position Manager</b> ブース タパック	さまざまな位置エンコーダとのインターフェイスを確立する、将来の評価基板。TIDM-1010 基板は Position Manager BoosterPack 評価基板と同じです (セクション 2.3.3 を参照)。
<b>PM</b>	Position Manager – ポジション エンコーダ インターフェイス用の C2000 デバイ스에搭載されている基本的なハードウェアおよびソフトウェア
<b>PM_bissc</b> また は <b>PM_BISSC</b>	すべてのライブラリ関数に使用される接頭辞。
<b>ポイント ツー ポ イントトポロジ</b>	1 つのデバイス (エンコーダまたはセンサ) のみがマスタと接続される構成。
<b>SSI</b>	シリアル同期インターフェイス
<b>SL</b>	BiSS-C 仕様に基づくものです。スレーブ リターン。スレーブ から BiSS-C マスタへのデータ入力。各 BiSS フレームで、位置および 1 ビットの CD データを含むエンコーダの応答です。
<b>SPI</b>	C2000 デバイス上の SPI (シリアル ペリフェラル インターフェイス) ペリフェラル。

## 8 著者について

**Lori Heustess** は、長年にわたって C2000 チームのメンバーを務めてきました。氏の関心分野は、CPU とペリフェラルの検証、ソフトウェア開発、産業用アプリケーションです。現在、特定用途向け MCU (ASM) 産業用アプリケーション チームで仕事をしています。

**Subrahmanya Bharathi Akondy** は、いくつかの C2000 MCU 製品と制御ペリフェラルのアーキテクチャ定義と設計に取り組んできました。氏の関心分野は、MCU のアーキテクチャ、アプリケーション、設計の側面です。

**Sheena Patel** は、C2000 MCU グループの産業用ドライブ チームで、プロダクト マーケティング エンジニアとして従事しています。

## 9 改訂履歴

### Changes from Revision \* (April 2018) to Revision A (November 2024)

Page

ドキュメント全体にわたって表、図、相互参照の採番方法を更新.....	1
LaunchPad 情報を更新.....	1
BiSS インターフェイス ウェブサイトへのリンクを追加。ライブラリの機能概要を追加.....	2
「システム概要」セクションに導入部を追加 .....	4
ブロック図を更新.....	4
「主な使用製品」セクションに導入部を追加 .....	4
「主な使用製品」セクションにデバイスへのハイパーリンクを追加 .....	4
MCU および LaunchPad の情報を F28P65x に変更.....	4
「設計上の考慮事項」セクションに導入部を追加 .....	5
「BiSS-C プロトコル」セクションを追加.....	5
「ライン遅延補償」セクションを追加.....	7
「エンコーダによる処理時間要求」セクションを追加 .....	8
「制御通信」セクションを追加.....	8
「C2000 BiSS-C エンコーダ インターフェイスの概要」セクションを追加 .....	9
機能概要を追加。TIDM-1010 基板および BOOSTXL-POSMGR コネクタの表で、特に TIDM-1010 による使用を示す第 3 列を更新。BoosterPack ブロック図を追加 .....	10
「MCU のリソース要件」セクションを追加.....	13
「入力、出力信号、CLB タイル」セクションを追加.....	13
「CLB BiSS-C 実装の詳細」セクションを追加.....	15
「トランザクションの波形」セクションを追加.....	15
「FRAME_STATE の生成」セクションを追加.....	17
「CLB_SPI_CLOCK の生成」セクションを追加.....	19
「ENCODER_CLOCK (MA) 生成」セクションを追加.....	21
「PM BiSS-C インターフェイス ライブラリ」セクションに導入部を追加 .....	23
『C2000 BiSS-C エンコーダ インターフェイスのソフトウェア ガイド』の詳細情報を削除。ソフトウェアガイドのハイパーリンクを追加。ライブラリ関数の概要を追加.....	24
「ハードウェア」セクションに導入部を追加 .....	25
TIDM-1010 基板のジャンパの詳細表を更新 (注 2 を更新、未使用の注 3 を削除) .....	26
『C2000 BiSS-C エンコーダ インターフェイスのソフトウェア ガイド』の概要とハイパーリンクを追加。ソフトウェアアーキテクチャの図および通信デモのフローチャートの図を追加.....	27
「C2000 ドライバ ライブラリ (DriverLib)」セクションを追加 .....	29
「C2000 SysConfig」セクションを追加 .....	30
「C2000 構成可能ロジック ブロック ツール」セクションを追加 .....	30
「CCS と controlSUITE のインストール」セクションを「Code Composer Studio™ と C2000WARE-MOTORCONTROL-SDK のインストール」セクションに変更 .....	30
「リファレンス ソフトウェアの場所」セクションを追加 .....	30
『C2000 BiSS-C エンコーダ インターフェイスのソフトウェア ガイド』の手順を削除。ソフトウェア ガイドの概要を追加.....	33
『C2000 BiSS-C エンコーダ インターフェイスのソフトウェア ガイド』の指示を変更し、アプリケーション例の概要を追加.....	33
テストされたエンコーダ表のデータを更新。ケーブル長の推奨事項を更新.....	33
「ベンチマーク」セクションを追加 .....	34
「トラブルシューティング」セクションを追加 .....	34
ハイパーリンクを更新.....	36
iCHaus、BiSS インターフェイス、Code Composer Studio、C28x Academy、C2000 ソフトウェア ガイドへのハイパーリンクを追加。『BiSS-C ソフトウェア ガイド』へのハイパーリンクを更新.....	36
API、BiSS フレーム、BiSS ライセンス、CD、CDM、CDS、CTS、MA、SLA、SL などの追加定義を追加.....	36

- 
- 著者のリストに Lori Heustess を追加.....37
-

## 重要なお知らせと免責事項

テキサス・インスツルメンツは、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、テキサス・インスツルメンツ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 テキサス・インスツルメンツ製品の選定、(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ます。

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 テキサス・インスツルメンツ製品を使用するアプリケーションの開発の目的でのみ、テキサス・インスツルメンツ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。テキサス・インスツルメンツ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、テキサス・インスツルメンツおよびその代理人を完全に補償するものとし、テキサス・インスツルメンツは一切の責任を拒否します。

テキサス・インスツルメンツの製品は、[テキサス・インスツルメンツの販売条件](#)、または [ti.com](https://www.ti.com) やかかる テキサス・インスツルメンツ製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。テキサス・インスツルメンツがこれらのリソースを提供することは、適用されるテキサス・インスツルメンツ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。

お客様がいかなる追加条項または代替条項を提案した場合でも、テキサス・インスツルメンツはそれらに異議を唱え、拒否します。

郵送先住所: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2024, Texas Instruments Incorporated



## 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ます。

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](#)、または [ti.com](#) やかかる TI 製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。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。

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

郵送先住所 : Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2024, Texas Instruments Incorporated