

## Design Guide: TIDM-1010

## 适用于 C2000™ MCU 的 BiSS-C 绝对编码器主接口参考设计



## 说明

C2000™ 微控制器 (MCU) Position Manager 技术可提供集成式解决方案来连接最常用的数字和模拟位置传感器，而无需外部现场可编程门阵列 (FPGA) 或应用特定集成电路 (ASIC)。Position Manager BoosterPack™ 是一种用于评估各种编码器接口的灵活且具有成本效益的平台，旨在与多个 C2000 LaunchPad™ 开发套件一起工作。该参考设计的软件专门用于实施 BiSS-C™ (位置编码器的数字双向接口)。该参考设计中包括高度优化且易于使用的软件库和示例，支持使用位置管理器 BoosterPack 来运行 BiSS-C 位置编码器。

## 资源

<a href="#">TIDM-1010</a>	设计文件夹
<a href="#">LAUNCHXL-F28P65X</a>	工具文件夹
<a href="#">SN65HVD78</a>	产品文件夹
<a href="#">TLV702</a>	产品文件夹
<a href="#">TPS22918-Q1</a>	产品文件夹

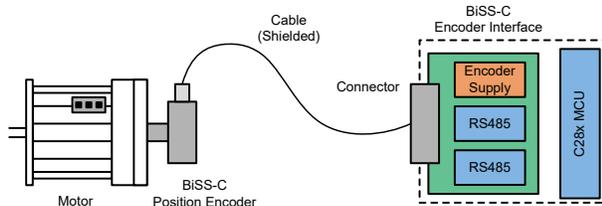
## 特性

- 用于位置编码器接口的灵活低电压 BoosterPack 评估平台
- 用于 BiSS-C 的集成 MCU 解决方案，无需额外的 FPGA
- 可通过库提供的驱动程序函数和数据结构轻松使用连接 BiSS-C 的诸项命令
- 用于解压缩接收数据的库支持和优化的循环冗余校验 (CRC) 算法
- 支持高达 10MHz 的时钟频率 (在长达 100m 的电缆长度运行条件下进行了验证)
- 包括展示 BiSS-C 软件库的评估软件示例

## 应用

- [工业](#)
- [电机驱动器](#)

请咨询 TI E2E™ 支持专家



## 1 系统说明

工业驱动器 (如何伺服驱动器) 需要精确、高度可靠且低延迟的位置反馈。BiSS 协议旨在实现传感器和控制器之间的数字数据串行传输。BiSS 表示双向串行同步。BiSS 接口由 iC-Haus GmbH 在 2002 年作为开源协议推出。BiSS-C 模式是连续模式，在该模式下，BiSS-C 接口循环读取位置数据。

TIDM-1010 设计实现了连接到 C2000 LaunchPad 的 BiSS-C 接口。BiSS-C 是基于 RS-485 标准的纯串行数字接口。BiSS 能够传输位置值以及其他物理量，并支持对编码器的内部存储器进行读取和写入。传输的数据类型包括绝对位置、转数、温度、参数、诊断等。图 1-1 展示了此设计中使用的主要硬件模块。

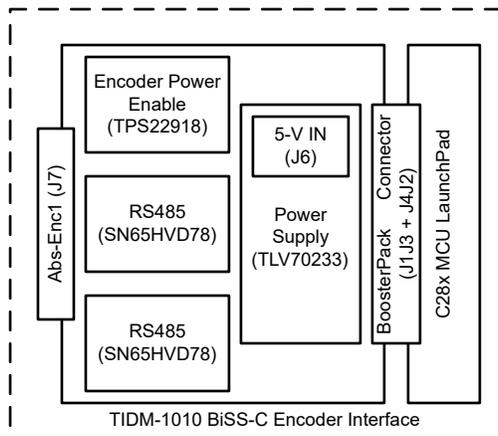


图 1-1. TIDM-1010 硬件模块和连接器

TIDM-1010 支持通常与 BiSS 位置或旋转编码器一起使用的点对点配置。图 1-2 中展示了点对点拓扑。在点对点配置中，主器件仅运行一个带有一个或多个传感器的器件。PM\_bissc 库仅支持此配置。

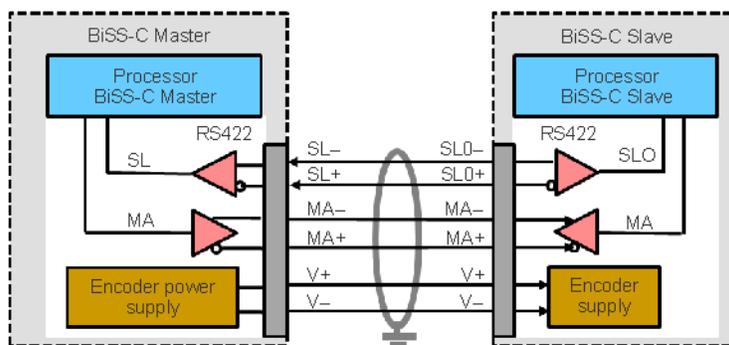


图 1-2. BiSS-C 点对点结构

绝对位置编码器通过 6 线屏蔽电缆连接到 TIDM-1010。RS-485 用作物理层。6 条线分别是：

- MA+ 和 MA-：编码器时钟 (BiSS 主器件时钟) 差分信号
- SL+ 和 SL-：编码器返回数据 (BiSS 从器件返回) 差分信号
- V+ 和 V-：编码器的电源和接地电源

BiSS-C MA 时钟频率是可变的，取决于电缆长度 (请参阅节 3.3.4)。最大电缆长度或可实现的最大时钟频率可能会有所不同，具体取决于编码器和编码器电缆。电缆的质量会影响通信性能，因此编码器制造商在其数据表中规定了这些限值，并建议使用适合其编码器的电缆。有关协议和点对点配置的更多详细信息，请访问 [BiSS 接口：BISS 用户协会和互联网平台](#)。

德州仪器 (TI) C2000 Position Manager BiSS-C (PM\_bissc) 编码器接口无需 FPGA 或 CPLD 等外部硬件即可支持 BiSS 实现。此参考设计具有以下特性：

- 高达 10MHz 时钟频率，电缆长度为 10m
- 集成电缆传播延迟补偿
- 软件驱动程序功能：
  - 执行与编码器的事务。这包括向编码器发送 MA 信号并接收响应。
  - 计算 CRC
  - 将接收到的 CRC 与计算得出的 CRC 进行比较
  - 解包响应数据

此参考设计包括所有源代码。用户可以根据其应用的需要进行实现所需的任何更改。

### 备注

该库支持基本接口驱动程序

- 单周期数据 ( 位置 + 错误 + 警告 + CRC ) 事务
- 单个寄存器读取访问
- 单个寄存器写入访问

用户可使用该实现提供的基本接口来开发所有更高级别的应用软件和 BiSS 功能。

## 1.1 主要系统规格

表 1-1. 主要系统规格

参数	规格	详细信息
输入电压	5V <sup>(1)</sup>	节 3.3.1
输出电压 ( 编码器 )	5V	节 3.3.1
支持的协议	BiSS-C 点对点	BiSS
最大频率	10MHz	支持长达 10m 的电缆。节 2.3.1.1
编码器位数	BiSS-C 协议标准	BiSS
位置数据 CRC	$x^6 + x + 1$	用于位置数据 ( 单周期数据 ) 验证的多项式
控制数据 CRC	$x^4 + x + 1$	用于控制数据验证的多项式
控制数据功能	单个寄存器读取, 单个寄存器写入	
CPU 周期	节 3.3.5	
代码大小	节 3.3.5	

- (1) 连接到 TIDM-1010 器件的编码器决定了该电源所需的电流限制。TI 建议使用具有可调电流限制的通用台式可调电源, 而不是 TIDM-1010 生成的电压。

## 2 系统概述

C2000 BiSS TIDM-1010 参考设计是硬件和软件的组合。核心硬件组件是 C2000 实时微控制器 (MCU) 和 RS-485 收发器。C2000 LaunchPad 和包含 RS-485 收发器的 TIDM-1010 BoosterPack 是此实现中使用的电路板。C2000Ware 电机控制 SDK 软件包包含必要的软件。该软件包括一个用于实现主要 BiSS 接口功能的库和一个用于演示 BiSS-C 通信的系统级示例。

BiSS 编码器接口采用 C2000 CLB (可配置逻辑块) 和 SPI (串行外设接口) 模块。CLB 控制 MA 时钟、SPI 时钟, 并补偿电缆传播延迟。SPI 模块充当 RS-485 物理层的接收接口。固件用 C 语言编写, 并在 C2000 MCU 的 C28x 上运行。

C2000 LaunchPad 可为 TIDM-1010 RS-485 收发器供电并为编码器提供 5V 电压。如果编码器规范要求的电流高于 LaunchPad 可提供的电流, 也可以单独提供 5V 电压。

在启动期间, C28x 上运行的应用程序会初始化 MCU 时钟并配置引脚多路复用器。MCU 的 SPI 和 CLB 也需要配置用于发送和接收数据。

### 2.1 方框图

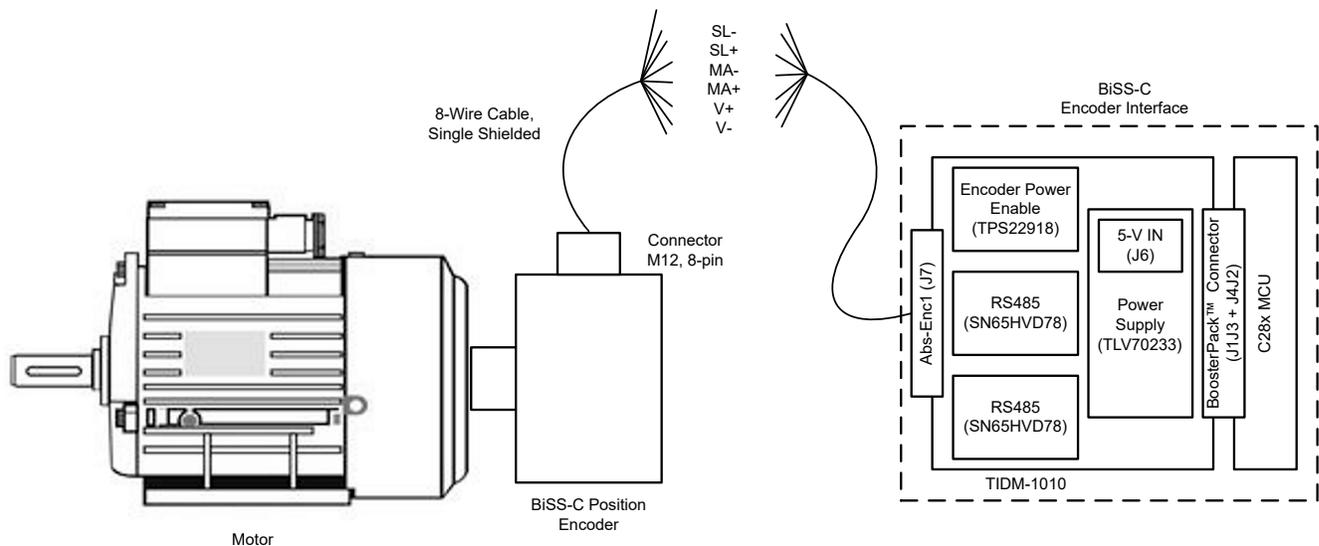


图 2-1. TIDM-1010 系统方框图

### 2.2 主要产品

TIDM-1011 参考设计硬件包含一个 C2000 LaunchPad 和一个 BOOSTXL-POSMGR BoosterPack。本节介绍使用的主要器件。有关这些器件中每一款的更多信息, 请参阅 TI.com 上相应的产品文件夹。

#### 2.2.1 C2000 实时 MCU LaunchPad

提供的 TIDM-1010 示例基于 TMS320F28P650DK9 MCU。F28P65x 在两个 200MHz C28x CPU 和实时控制协处理器 (CLA) 之间提供 600MIPS 的总体系统性能。这款强大的 MCU 包含 1MB 的内部 FLASH 以及独特性能的外设, 例如 16 位或 12 位模数转换器 (ADC)、比较器、12 位数模转换器 (DAC)、 $\Delta$ - $\Sigma$  sinc 滤波器、HRPWM、eCAP、eQEP 和 CAN 等。

F28P65x 还具有可配置逻辑块 (CLB)。BiSS 接口广泛使用 CLB。CLB 外设允许用户在无需外部 FPGA 或 CLPD 的情况下整合自定义逻辑。CLB 由组合在一起以启用自定义数字逻辑的子模块组成。子模块包括: 有限状态机 (FSM)、查找表 (LUT) 和计数器。CLB 还与现有片上控制外设连接, 以增强功能并提供设计选项。BiSS 接口可以移植到任何包含 CLB 的 C2000 器件。

LAUNCHXL-F28P65X 评估板用于 TIDM-1010 实现。LAUNCHXL-F28P65X LaunchPad 是适用于 F28P65x 器件的低成本开发板。

### 2.2.2 SN65HVD78

SN65HVD78 器件将一个差分驱动器和一个差分接收器组合在一起，这两个器件由一个 3.3V 单电源供电。驱动器差分输出和接收器差分输入在内部进行连接，形成适用于半双工（双线总线）通信的总线端口。这些器件都具有宽共模电压范围，因而适用于长线缆上的多点应用。

请在 [SN65HVD78](#) 产品文件夹中查找完整的器件特性和规格。

### 2.2.3 TLV702

TLV702 系列低压降 (LDO) 线性稳压器是具有出色线路和负载瞬态性能的低静态电流器件。所有器件版本具有热关断和电流限值以保证安全。这些器件在没有负载的情况下可调节至指定的精度。

请在 [TLV702](#) 产品文件夹中查找完整的器件特性和规格。

### 2.2.4 TPS22918-Q1

TPS22918-Q1 是一款单通道负载开关，可对上升时间和快速输出放电进行配置。该器件包含一个可支持 2A 最大连续电流的 N 沟道 MOSFET。开关由一个开关输入控制，能够直接连接低电压控制信号。

请在 [TPS22918-Q1](#) 产品文件夹中查找完整的器件特性和规格。

## 2.3 设计注意事项

本节提供以下方面的内容：

1. BiSS-C 接口协议概述。
2. C2000 BiSS-C 编码器接口概述。
3. TIDM-1010 硬件 (BOOSTXL-POSMGR BoosterPack) 实现。
4. C2000 MCU 实现，包括所需的输入/输出、CRC 计算和 CLB 设计。
5. C2000 BiSS-C 编码器接口软件库概述。

### 备注

本节仅提供实现细节。有关以下方面的信息：

- 硬件要求、设置和测试：请参阅 [节 3](#)
- 软件：安装和运行软件：参阅：[C2000 BiSS-C 编码器接口软件指南 \(HTML、PDF\)](#)。该软件指南包括以下文档：
  - 通信演示
  - BiSS-C 应用程序编程接口 (API)
  - 将库整合到您自己的解决方案中
  - 将库移植到双核器件上的 CPU2

### 2.3.1 BiSS-C 协议

BiSS-C 接口规格介绍了与编码器和其他传感器进行通信的串行接口协议。BiSS-C 允许通过同一条线路同时传输位置数据和控制数据。该接口类似于串行同步接口 (SSI) 协议，其中的数据传输与控制器时钟信号同步。

表 2-1. SSI 与 BiSS-C 的比较

SSI ( 串行同步接口 )	BiSS-C
2Mbits/s	高达 10Mbits/s ( 在 10 米电缆上 )
不适用	电缆长度传播补偿
差分双绞线电缆 (RS-422/485)	差分双绞线电缆 (RS-422/485)
单向。仅支持从编码器到控制器的数据传输。	通常单向使用，如 SSI。还支持双向数据传输，使编码器配置可由驱动器控制。
支持简单的奇偶校验	通过循环冗余校验 (CRC) 实现稳健错误检查
不适用	编码器可请求额外的处理时间
仅点对点拓扑	通常是点对点的，但该协议也支持菊花链式编码器。本文档中讨论的 C2000 设计实现了点对点拓扑。

### 备注

本文档介绍了 BiSS-C 帧和控制通信的一些重要概念。有关详细规格，请参阅 [BiSS 用户协会和互联网平台](#)

尽管位置编码器通常提供电机的反馈位置数据，但该编码器允许对电机进行闭环控制。此设计实现了用于点对点通信的 BiSS-C 接口。在点对点配置中，控制器或驱动器仅连接一个这样的编码器。

在 BiSS-C 协议中，控制器向编码器提供时钟 (MA)，然后编码器将数据发送回控制器 (SL)。通信数据包括单周期数据 (SCD) 和控制数据 (CD)。

- SCD 是从编码器发送到控制器的主器件数据，每个 BiSS 周期更新一次。SCD 包括绝对位置、错误位、警告位和 CRC 校验。绝对位置可以是仅限单圈（旋转一周）的数据，也可包括单圈和多圈（旋转多周）数据。
- CD 使控制器能够写入和读取编码器寄存器。控制器将一个控制数据位 (CDM) 发送到编码器，编码器在每个周期将一个控制数据位 (CDS) 返回控制器。

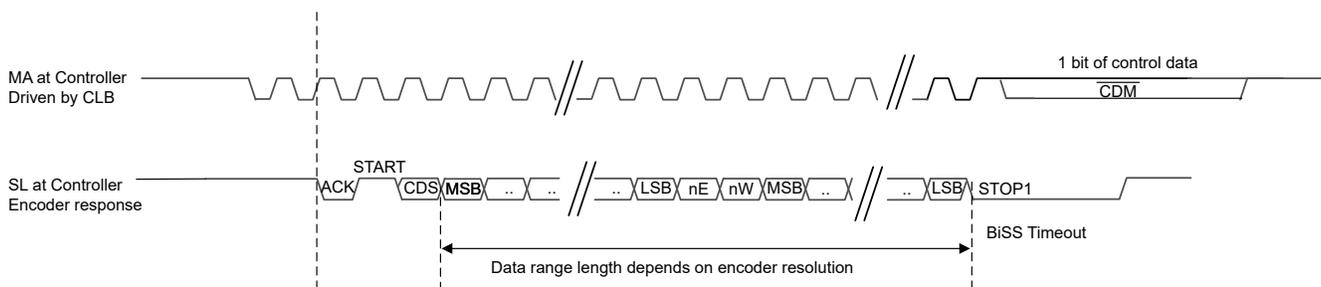


图 2-2. BiSS-C 帧

图 2-2 展示了一个示例 BiSS 帧。在复位状态下，MA 和 SL 线路都处于高电平有效状态。控制器通过在 MA 线路上发送时钟来开始该帧。在 MA 时钟的第二个上升沿，编码器以低电平信号响应，以确认 (ACK) BiSS 帧。在下一个 MA 时钟周期中，编码器将开始位置为有效。开始之后，编码器发送控制数据 (CDS) 位，用于响应前一帧中发送的控制 CDM 位。

在 CDS 位之后，编码器从最高有效位 (MSB) 开始发送位置数据。位置数据后跟一个错误位 (nE) 和一个警告位 (nW)。然后，编码器发送循环冗余校验 (CRC) 位，MSB 优先。编码器在 SL 线路上反相发送 CRC。

发送所有位之后，编码器进入 BiSS 超时状态，将 SL 驱动为低信号电平。当编码器为下一次传输做好准备或 BiSS 超时到期时，SL 变为高电平。在 BiSS 超时期间，MA 时钟线路的反转状态是用于控制通信的 CDM 位。控制器在每个 BiSS 帧中发送一个控制数据位，编码器发送回一个位。

### 备注

编码器具有一个 SLI (输入) 和一个 SLO (输出) 信号。在点对点配置中，SLO 信号直接连接到控制器的 SL。

#### 2.3.1.1 线路延迟补偿

在实际应用环境中，编码器可以远离控制器。编码器和控制器之间的长电缆连接可能会延迟传输和物理噪声。

线路延迟是由于传输所用电缆的长度而导致的传播延迟。当控制器开始发送 MA 时钟时，时钟需要一些时间才能到达编码器。当编码器接收到时钟时，编码器开始用 SL 数据进行响应。编码器的响应也会通过电缆沿相反的路径传输到控制器。通过线缆传输数据的延时时间与电缆的长度成正比。对于长达 100 米的电缆，从控制器发送时钟的时间到控制器接收到编码器响应，1 $\mu$ s 的电缆延迟是可能的。

BiSS-C 接口具有补偿线路延迟的机制，可避免在较长电缆传输时出现错误。

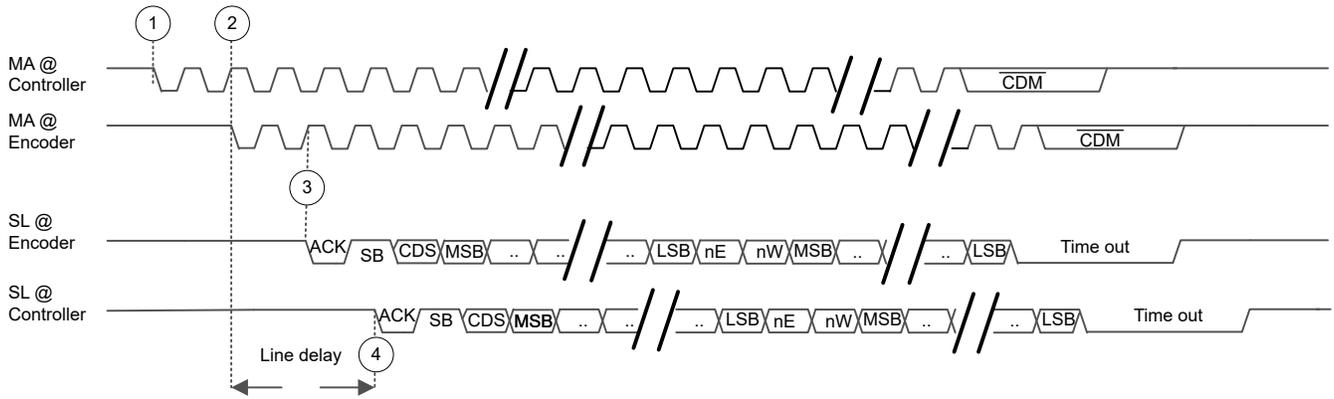


图 2-3. BiSS-C 线路延迟

图 2-3 从两个角度展示信号：控制器处和编码器处。

请参阅图 2-3 标记 (1)：

控制器线路上的 MA 显示了时钟在 BiSS-C 接口上的外观。这位于驱动控制器所在的冷侧。控制器通过发送 MA 信号开始事务。

请参阅图 2-3 标记 (2)：

由于线路延迟，MA 时钟信号在编码器（电机）处延迟。编码器线路上的 MA 显示延迟。

请参阅图 2-3 标记 (3)：

编码器响应延迟 MA 时钟的第二个上升沿。编码器线路的 SLO 显示编码器对编码器处 MA 的响应。

请参阅图 2-3 标记 (4)：

响应需要一些时间才能传回控制器。如控制器信号处的 SLO 所示，到达控制器的时间会延迟。通过测量 MA 时钟的第二个上升沿与 SLO 线路的第一个下降沿之间的持续时间，可计算总延迟时间。为了避免传输错误，BiSS-C 接口会补偿这种线路延迟。

### 2.3.1.2 编码器处理时间请求

编码器可以在发送传感器数据之前请求处理时间。模数转换和存储器访问等操作需要额外的时间。编码器通过延迟开始位 (SB) 来指示处理时间。控制器必须检查编码器是否请求处理时间并提供额外的时钟周期。

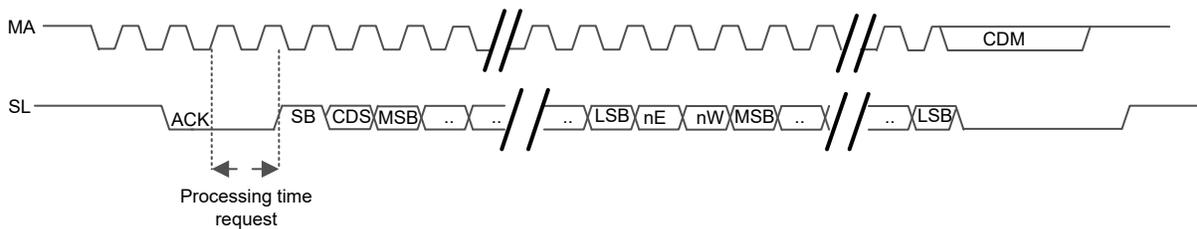


图 2-4. 编码器处理时间请求

### 2.3.1.3 控制通信

在 BiSS-C 通信中，控制器可以通过 MA 线路发送控制帧而不会中断位置数据通信。这是通过在每个 BiSS 帧内发送控制帧的一个位来实现的。

如节 2.3.1 中介绍的，控制器每个 BiSS 帧发送一个数据位，称为 CDM。同样，编码器每个 BiSS 帧使用一位响应（称为 CDS）来响应这些 CDM 位。这会重复进行，直到通过多个 BiSS 帧发送完整的控制帧并收到响应。

有两种类型的 BiSS-C 控制帧：

- 寄存器通信帧：编码器中内部寄存器的读取或写入
- 命令帧：向编码器发送命令

**备注**

根据提供的信息，TIDM-1010 不实现命令帧。如果系统开发人员需要，可将此功能添加到设计中。在命令帧中，控制选择位 (CTS) 为零 (CTS = 0)。使用命令帧来支持多点连接不在此设计的范围内。因此，本文档仅关注寄存器通信帧。

以下步骤介绍了读取或写入访问。请参阅图 2-5 和图 2-6。

1. 控制器在 CDM = 0 时发送至少 14 个 BiSS-C 帧
2. CDM = 1 指示控制帧的开始位 S。
3. 下一个 CDM 位称为 CTS (控制选择位)。对于寄存器访问，CTS 为 1。
4. 然后，控制器发送一个 3 位 ID，以识别正在访问的从器件。
5. ID 后跟一个 7 位寄存器地址和一个 CRC。
6. 接下来的 3 个位为读取位 (R)、写入位 (W) 和开始位 (S)。R W S 定义如下：
  - 写入访问：RWS 等于 011b
  - 读取访问：RWS 等于 101b
7. 控制器：
  - 将 CDM 位保持为低电平，以进行读取访问
  - 发送要写入的 8 位数据 + CRC，以进行写入访问
8. 停止位 (P) 指示控制帧结束。

**备注**

在第 6 步中，该协议允许编码器为读取或写入请求额外的处理时间。这是通过用 S = 0 (而不是如图所示的 S = 1) 进行响应来完成的。当前实现不支持此功能，但开发人员可以通过更新 CD 状态机 C 代码来添加此功能。并非所有编码器都需要此额外时间。请参阅特定编码器的规格。

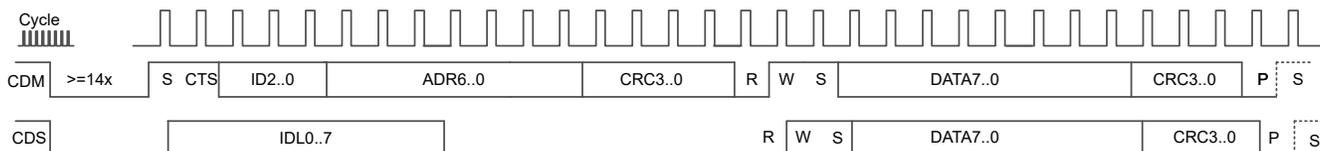


图 2-5. 控制帧：寄存器读取

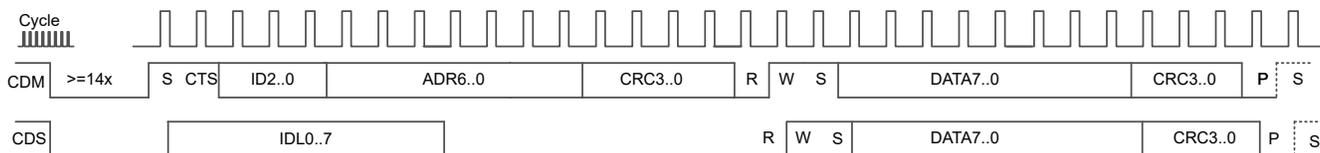


图 2-6. 控制帧：寄存器写入

**备注**

BiSS 协议支持连续寄存器的背对背读取或背对背写入。当控制器发送一个停止位 (P = 1) 并立即后跟另一个开始位 (S = 1) 时，此功能启用。此功能未在 TIDM-1010 中实现。每个控制帧仅支持单次读取或单次写入。

### 2.3.2 C2000 BiSS-C 编码器接口概述

通过 BiSS-C 编码器接口进行的通信主要由以下组件实现：

- CPU (C28x)
  - 配置器件、CLB 和 SPI
  - 初始化 CLB 计数器，以便为编码器分辨率生成适当的 MA 时钟频率和时钟计数
  - 打包和解包数据
  - 计算单周期数据 CRC 和控制帧 CRC
  - 将计算出的 CRC 与接收到的 CRC 进行比较

- 可配置逻辑块 (CLB)
  - 发送 MA 时钟和 CDM 位
  - 监控 SPI PICO 信号以获取编码器的响应。控制 SPI 时钟以读取响应
  - 根据接口的要求，测量和补偿电缆传播延迟
- 串行外设接口 (SPI)
  - 接收编码器的响应
- 器件互连 (XBAR、CLB XBAR)
  - 路由进出 CLB 和器件的信号
- 外部接口块
  - 带有 RS-485 差分线路驱动器的 TIDM-1010 板

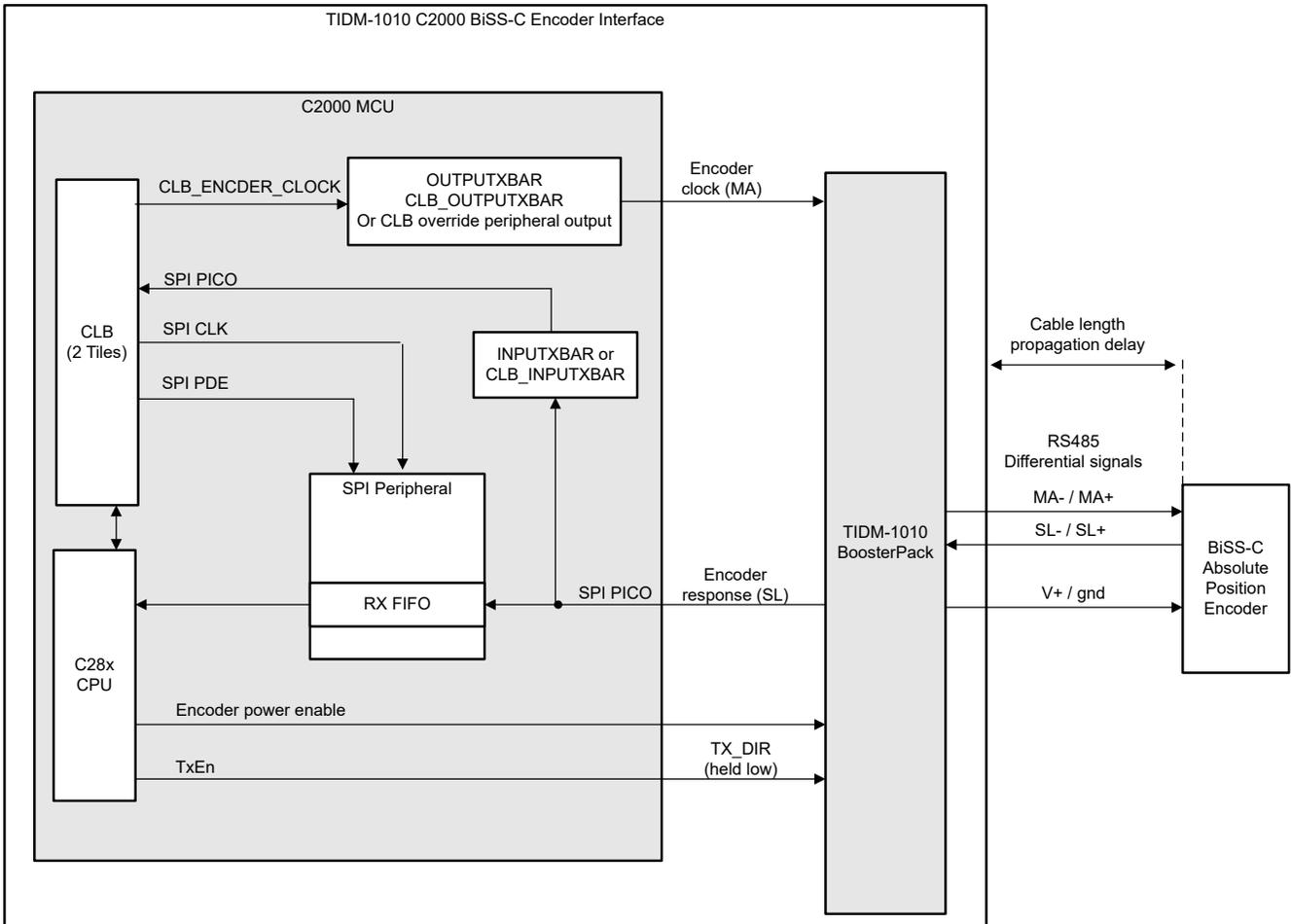


图 2-7. 编码器接口实现方框图

**备注**

在 F2837xD/F2837xS/F28007 器件上，CLB 无法直接覆盖 SPI 输入信号。TIDM-1010 硬件可将 CLB 生成的 SPI 时钟路由到回外设时钟输入引脚，并将 SPI PDE 引脚接地。有关更多信息，请参阅 TIDM-1010 原理图。

本节的其余部分介绍了设计的以下方面：

- TIDM-1010 硬件
- 包括 CLB 的 C2000 MCU 资源
- 编码器接口使用的 C2000 软件

### 2.3.3 TIDM-1010 板实现

TIDM-1010 板提供以下功能：

- 用于 C2000 MCU 和编码器之间 RS-485 通信的差分线路驱动器和接收器。
- 从 MCU 路由到 RS-485 驱动器/接收器方向控制的 TxEN 信号。对于 BiSS-C 实现，该信号保持低电平。
- 路由到 GPIO 的 SPICLK 信号，在其中可由 CLB 外设控制。对于除 F2837xD、F2837xS 和 F28007x 器件之外的所有受支持器件，此连接是可选的。在其他器件上，CLB 能够为器件本身的 SPI 外设提供时钟。

#### 备注

TIDM-1010 板与 Position Manager BoosterPack (BOOSTXL-POSMGR) 相同，这意味着 TIDM-1010 板可以与其他几种类型的位置编码器相连接。默认情况下，该板兼容所有功能。该参考设计主要针对 BiSS-C，可以忽略本文档中未提及的硬件模块。

表 2-2 列出了 TIDM-1010 BiSS-C 实现上的连接器以及这些连接器的功能。

**表 2-2. TIDM-1010 板和 BOOSTXL-POSMGR 连接器**

连接器	说明	由 TIDM-1010 使用
Abs-Enc-1 (J7)	BiSS-C 和其他绝对编码器	是，LaunchPad 站点 2
Abs-Enc-2 (J8)	BiSS-C 和其他绝对编码器	否
Abs-Enc-2 破孔 (J10)	允许在站点 2 使用两个绝对编码器 (使用跳线)	否
SinCos (J14)	SinCos 编码器	否
旋转变压器 (J14 和 J15)	具有 15V 激励电路的旋转变压器接口	否
PTO (J17)	脉冲序列输出	否
J1、J3 和 J4、J2	BoosterPack 连接器	是
J6	5V 直流电源输入	是
J16	15V 直流旋转变压器激励输入	否

图 2-8 介绍了 LaunchPad 开发套件每个站点上的编码器支持。

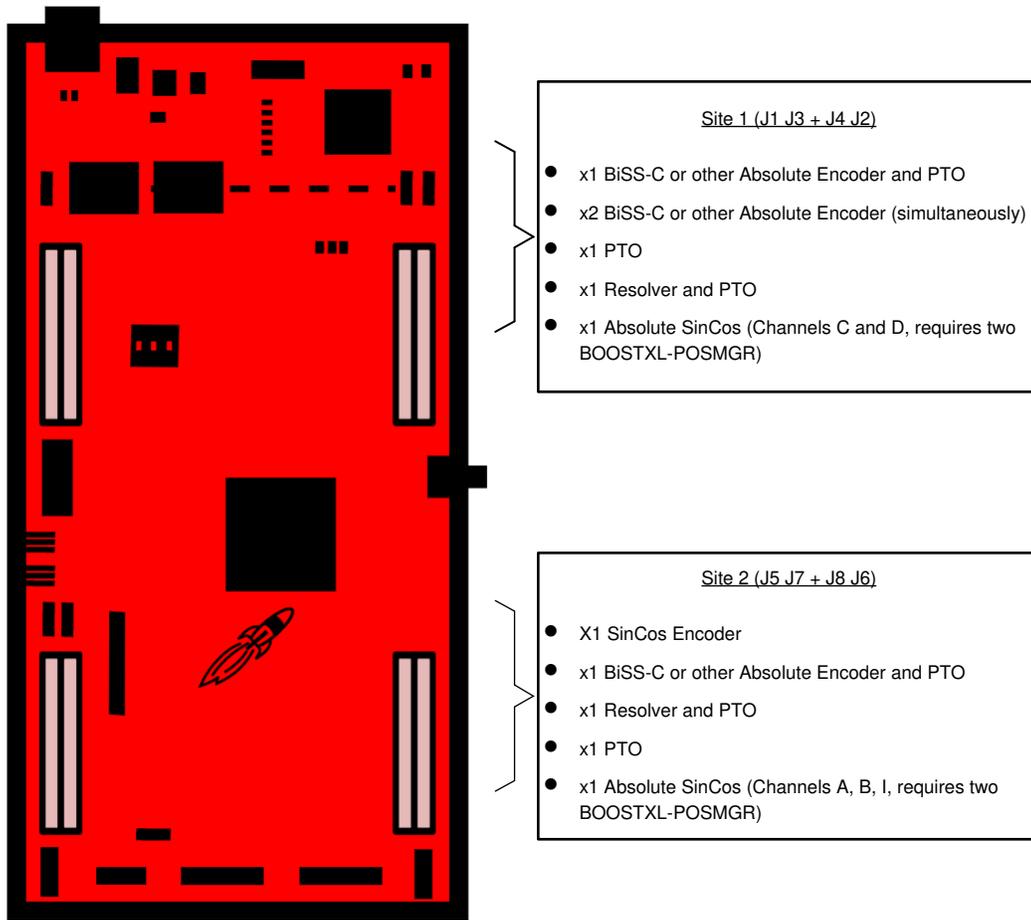


图 2-8. TIDM-1010 板和 BOOSTXL-POSMGR 编码器支持

根据提供的信息，TIDM-1010 使用 LaunchPad 站点 2 和 BOOSTXL-POSMGR 的编码器 1 连接。图 2-9 显示了相关连接。TIDM-1010 BoosterPack 的完整原理图可从 BOOSTXL-POSMGR 产品页面下载。

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

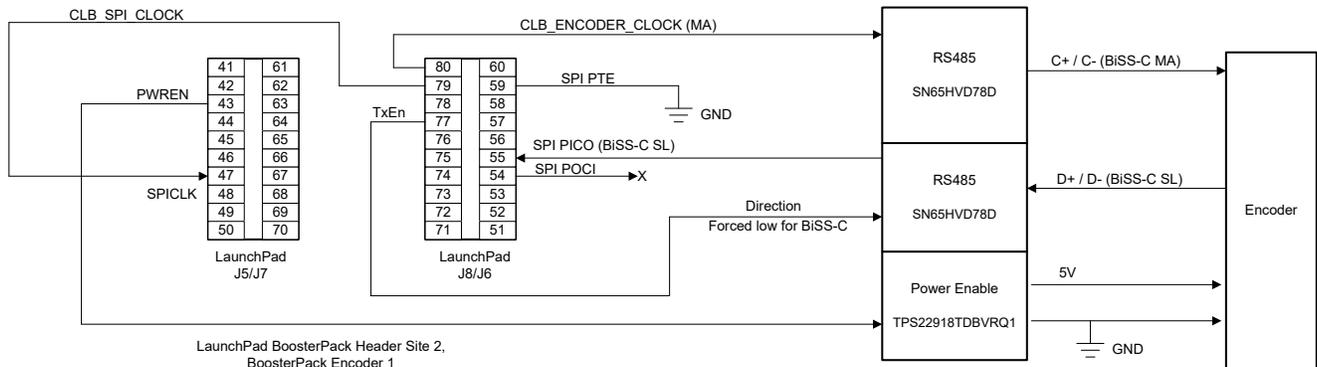


图 2-9. BoosterPack 方框图

### 2.3.4 MCU 资源要求

表 2-3 列出了 TIDM-1010 参考设计使用的 C2000 实时 MCU 资源。

**表 2-3. TIDM-1010 资源使用情况**

资源名称和数量	类型	用途
CLB x 2	类型 1 或更高版本	提供 SPI 时钟、延迟补偿和 CDM 位控制。如果逻辑块实例已更改，则还必须更新 CLB 的输入/输出路由。
GPIO x 3	I/O	<ul style="list-style-type: none"> <li>CLB 输出，MA 编码器时钟</li> <li>CLB 输出，RS-485 方向控制 (TxEN) 对于 BiSS-C，该信号保持低电平。</li> <li>CPU 编码器功率控制 (PwrCtl) 如果以其他方式控制编码器电源 (例如：外部供电)，则不需要该信号。</li> </ul>
GPIO x 1	I/O (仅限 F2837xD、F2837xS 和 F28007x)	<ul style="list-style-type: none"> <li>CLB_SPI_CLK 的 CLB 输出</li> <li>CLB 类型 1：将此引脚从外部路由到 SPICLK 输入。</li> <li>CLB 类型 2 或更高版本：直接从 CLB 为 SPI 模块提供时钟。无需外部连接，但可用于测试和调试。</li> </ul>
INPUTXBAR 或 CLB_INPUTXBAR x 1	模块、I/O	将 SPI PICO 引脚连接到 CLB 输入。
OUTPUTXBAR 或 CLB_OUTPUTXBAR 或外设输出的 CLB 覆盖 x 2	模块、I/O	<ul style="list-style-type: none"> <li>将 CLB 输出连接到 ENCODER_CLOCK (MA) GPIO</li> <li>将 CLB 连接到 TxEN GPIO</li> </ul>
SPI x 1	模块和 I/O	一个用于接收 RS-485 物理层数据信号的 SPI 实例。SPI 时钟由 CLB 控制。
CPU 和内存	模块	各种功能的 CPU 和内存使用情况。

#### 2.3.4.1 输入、输出信号和 CLB 逻辑块

本节介绍了 P65x 器件上使用的输入或输出和 CLB 逻辑块连接。当移植到另一个器件时，有时需要不同的路由或信号用法。

- 所使用的特定 GPIO 引脚和 SPI 模块取决于特定于器件的 LaunchPad 引脚排列。
- CLB 的输入和输出连接取决于该器件的特性以及引脚。例如，使用器件 INPUTXBAR 而不是 CLB\_INPUTXBAR。
- 特定的 CLB 逻辑块实例取决于逻辑块覆盖 SPICLK 等其他信号的能力。例如，如果使用 SPI-D，则最好在可直接访问 SPI-D 的逻辑块上实现设计。

#### 备注

在输入/输出图中：

- 彩色圆圈中的字母表示下一节中 CLB 逻辑块图内的分页连接。
- 后跟 **\_RE** 的字母表示 *上升沿*。例如：**D** 是编码器时钟。**D\_RE** 是施加 CLB 上升沿滤波器之后的相同信号。
- G** 是来自有限状态机的输出，称为 **FRAME\_STATE**。**G** 包含两个状态信号：**G.s0** 和 **G.s1**。

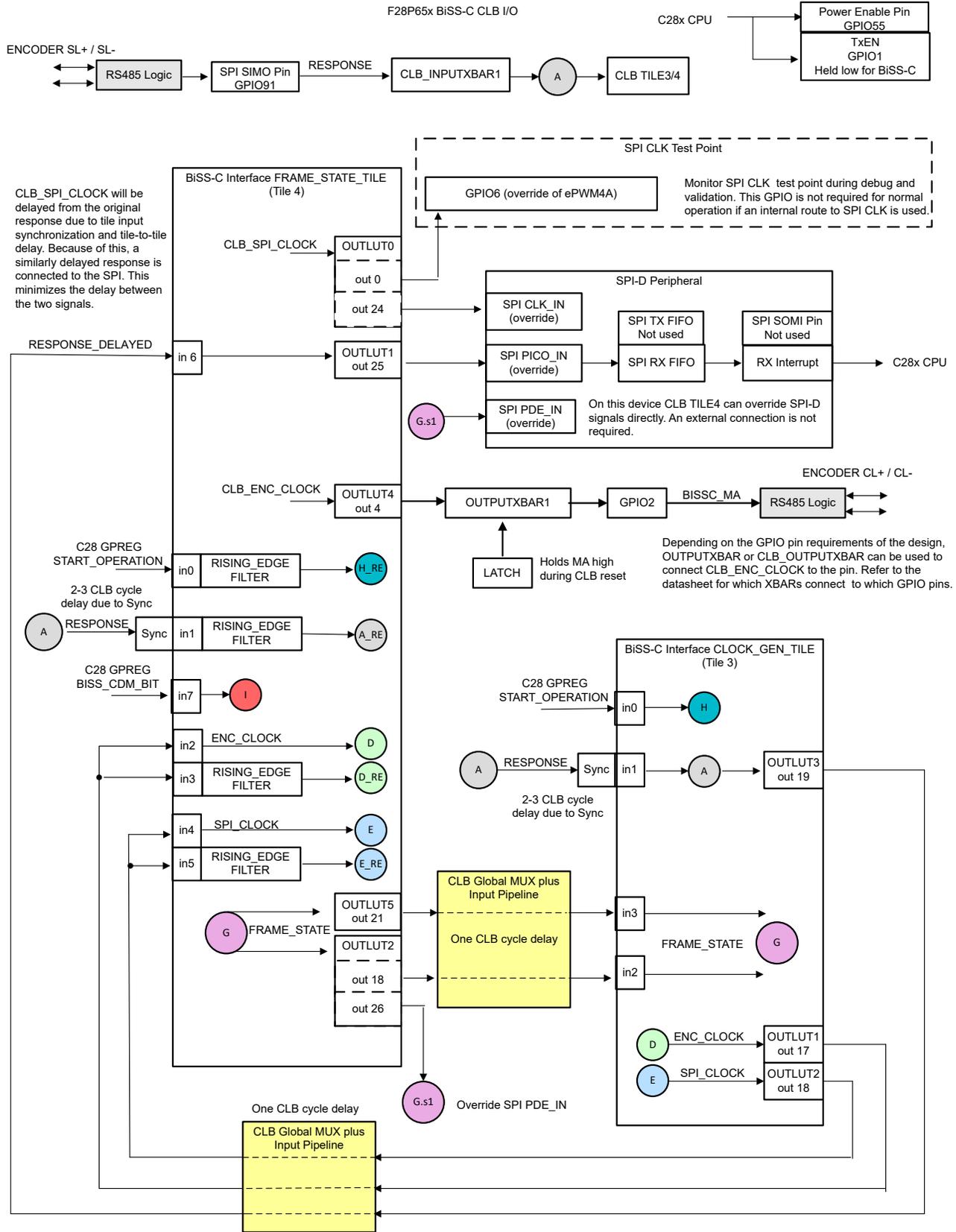


图 2-10. BOOSTXL\_POSMGR 的 F28P65 输入、输出和 CLB 用法

### 2.3.5 CLB BiSS-C 实现详细信息

CLB 负责以下工作：

- 使用 CDM 位生成编码器时钟 MA 信号。
- 监控 SPI PICO 信号以获得编码器响应。
- 将 SPICLK 与传入响应对齐。
- 为 SPI 提供时钟以接收响应。

本节介绍了使用两种方法设计 CLB 逻辑块：

- 使用波形显示事务在每个阶段的 CLB 行为。
- 包括子模块互连的 CLB 逻辑块设计。

#### 2.3.5.1 事务波形

在实现 CLB 设计时，首先使用波形直观地显示所需的 CLB 行为可能会有所帮助。为此，首先考虑一个示例事务。回想一下，BiSS-C 事务包含 MA 信号和编码器的响应。传输可以拆分成 FRAME\_STATE，如图 2-11 所示。第一步是将事务的每个元素映射到 CLB 子模块。表 2-4 展示了一个示例映射。

表 2-4. BiSS-C 事务到 CLB 映射

事务行为	CLB 映射
跟踪 FRAME_STATE	有限状态机 (FSM)：在给定的先前状态和当前输入的情况下转换到新状态。
生成两个特定宽度的时钟信号：一个用于 MA，一个用于 SPI 时钟。	对于每个信号，此要求会映射到计数器。利用匹配值来生成上升/下降沿的时序。然后，LUT (查找表) 会根据该时序生成实际的边沿。
为 MA 和 SPI 时钟传输特定数量的时钟脉冲。	对于每个时钟，这会映射到计数器。在时钟的边沿，计数器递增，MATCH 指示何时达到数量。
将 SPI 时钟与编码器的响应对齐	生成 SPI 时钟的计数器可以进行配置，使得边沿转换与编码器的响应正确对齐。
仅在接收到编码器的响应时允许为 SPI 提供时钟	LUT 会在不需要时钟时阻止时钟。
在 BiSS 超时期间，将 MA 信号保持为高电平或低电平 (CDM 位)	使用 GPREG 输入，C28 可向 CLB 指示 CDM 必须为高电平还是低电平。这与 LUT 组合，然后在传输特定数量的时钟后将值保持在 MA 上。
控制 TxEN	对于 BiSS-C，TxEN 保持低电平。CLB 输出 LUT 可对输出应用常量“0”。
通过发送 MA，指示 CLB 开始事务	C28x 配置用于事务的计数器和 SPI。CLB GPREG 允许 C28x CPU 直接更改 CLB 输入以开始事务。

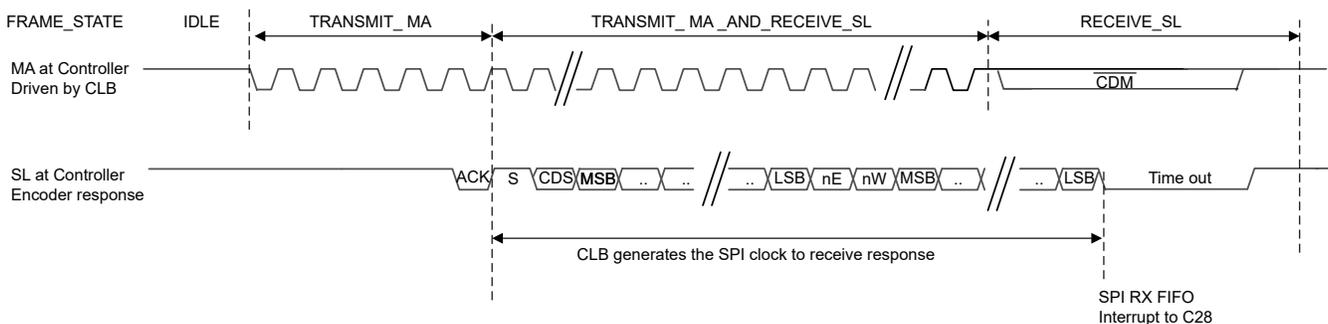


图 2-11. BiSS-C 单周期数据事务示例

下一步是实现特定子模块行为的可视化。从快速草图开始，然后在设计开发过程中添加其他详细信息。图 2-12 展示了使用 CLB SystemC 仿真模型生成的波形示例，该模型使用自定义输入作为编码器的响应。虽然图 2-12 是由实际设计生成的，但图的细节与初步草图类似。

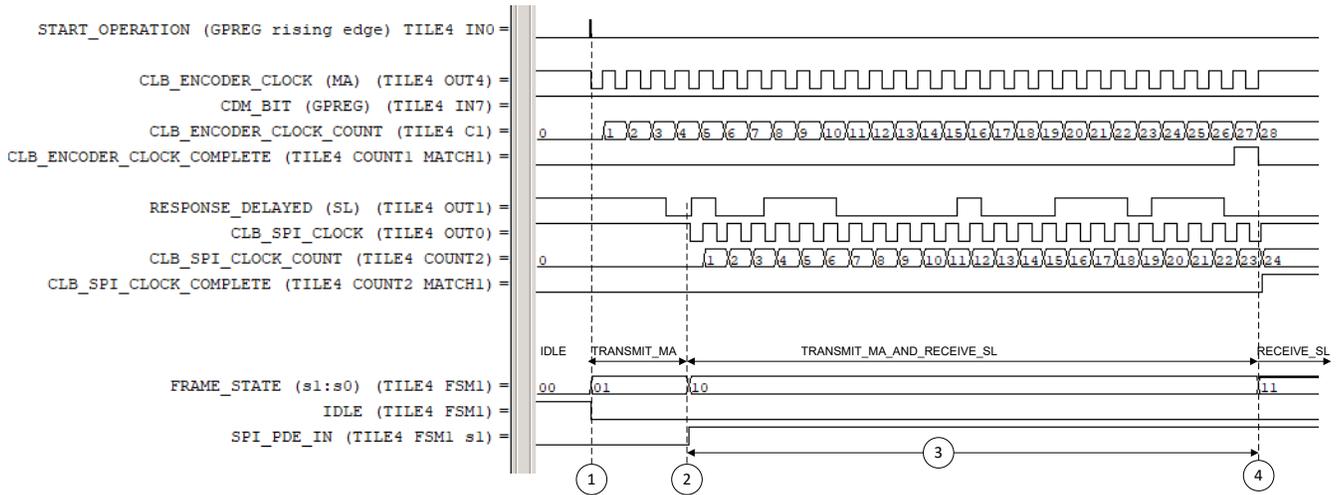


图 2-12. CLB 通信波形

以下部分中使用了 图 2-12 中的标记 1 至 4 来描述设计中与该标记相对应的特定行为。标记为：

1. 从 IDLE 转换到 TRANSMIT\_MA
2. 从 TRANSMIT\_MA 转换到 TRANSMIT\_MA\_AND\_RECEIVE\_SL
3. 在 TRANSMIT\_MA\_AND\_RECEIVE\_SL 期间
4. 转换到 RECEIVE\_SL

在 IDLE 状态期间，接口上没有任何活动。C28x 必须通过以下方式发起事务：

- 配置 CLB 和 SPI
- 通过 CLB 逻辑块的 GPREG 寄存器将 START\_OPERATION 信号拉高。GPREG 是 CLB 的通用寄存器，允许 C28x 直接控制逻辑块的输入。

参阅：图 2-12，标记 (1)。

START\_OPERATION 会在一个 CLB\_CLOCK 内保持高电平，因为针对该输入启用了 CLB 的上升沿滤波器。在标记 (1) 处，主状态机 (逻辑块 4 上的 FSM1) 会通过将 FRAME\_STATE 从 IDLE 移到 TRANSMIT\_MA 状态进行响应。

在 TRANSMIT\_MA 期间：编码器接口向编码器发送 MA 信号。此信号是具有特定占空比和所定义数量时钟周期的时钟。周期数取决于编码器的分辨率。在为 MA 提供时钟时，CLB 监控 SL 线路以获得编码器响应。

参阅：图 2-12，标记 (2)。

为了检测编码器的响应，CLB 监控 SL 的上升沿。上升沿对应于 ACK 之后的起始位。检测响应所需的时间可以是任意数量的时钟，具体取决于编码器的状态和电缆长度。此外，如果编码器需要额外的响应时间，则延长 ACK 状态。

检测到编码器的起始位时，FRAME\_STATE 转换用于发送 MA 和接收 SL。

参阅：图 2-12，标记 (3)。

CLB 开始为 SPI 外设提供时钟以接收响应。

- CLB 会将 CLB\_SPI\_CLOCK 与响应对齐。在 CLB 生成的时钟的上升沿对响应进行采样。
- 通过直接覆盖外设的输入来将 CLB\_SPI\_CLOCK 连接到 SPI\_CLK\_IN。
- 计数器会跟踪所生成的 SPI 时钟数量。此计数器的 Match1 指示何时发送了所需的 SPI 时钟。有时需要超出响应长度要求的额外 SPI 时钟，以将 SPI 的 FIFO 填充到 FIFO 中断级别。

参阅：图 2-12，标记 (4)。

### 2.3.5.2 FRAME\_STATE 生成

本节深入讨论了 FRAME\_STATE 生成。FRAME\_STATE 负责确定在事务中的任意给定时间生成哪些时钟。

FRAME\_STATE ( FSM\_1 s1、s0 ) 在 4 种状态之间转换：

- 空闲：
  - 无活动
  - CLB 等待 START\_OPERATION 信号的上升沿开始传输编码器时钟。
- TRANSMIT\_MA：
  - 传输编码器时钟
  - 监控响应以获得编码器 ACK
- TRANSMIT\_MA\_AND\_RECEIVE\_SL
  - 继续传输编码器时钟
  - 开始 SPI 时钟以接收响应
  - 在此期间，编码器时钟和 SPI 时钟均处于活动状态。
- RECEIVE\_SL：
  - 编码器时钟的传输完成
  - CDM 位在 MA 信号上处于活动状态
  - 通过为 SPI 提供时钟完成对响应的接收。
  - 根据数据长度和所使用的 SPI 宽度，可以生成额外的 SPI 时钟来触发 SPI 中断。

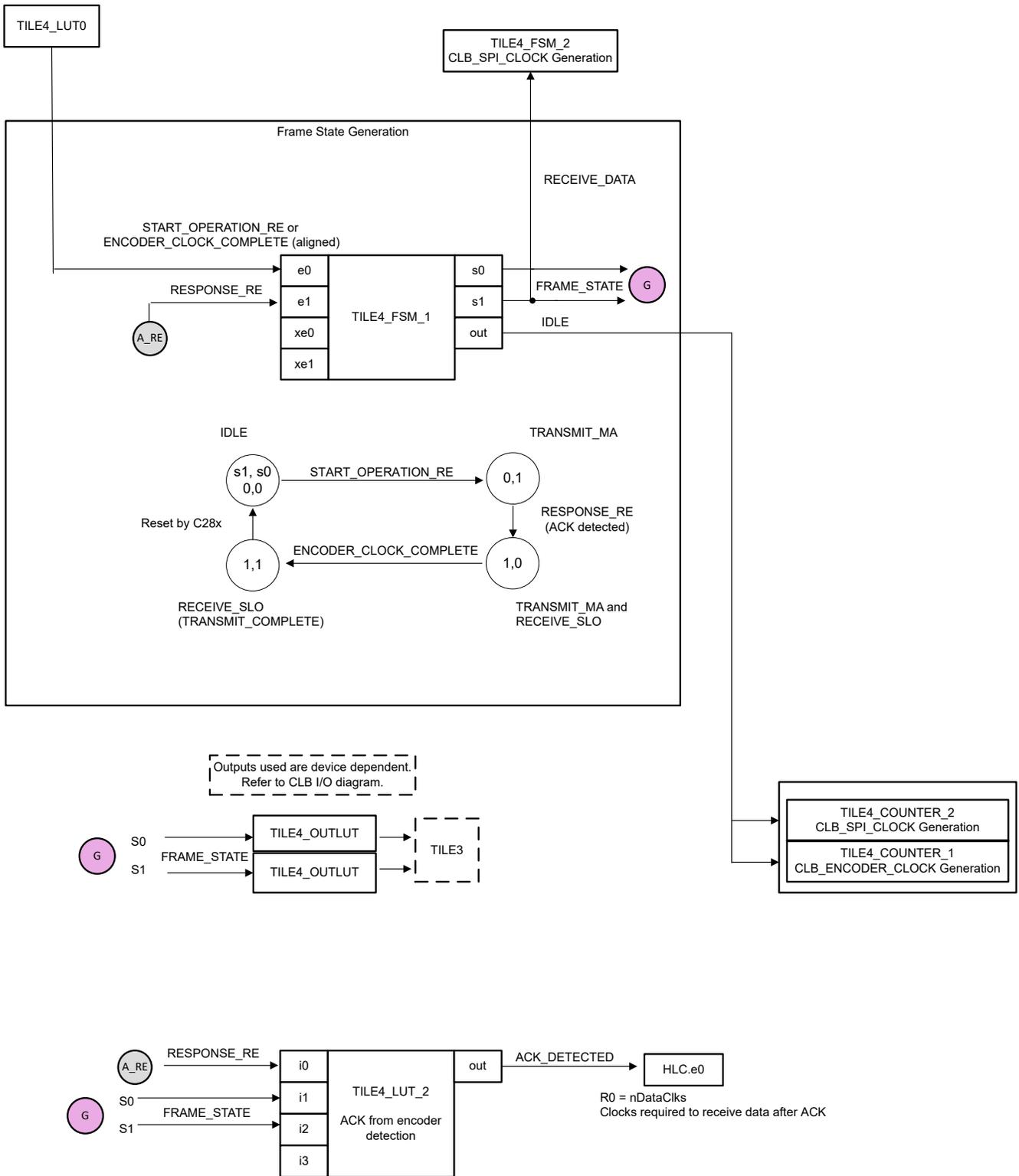


图 2-13. FRAME\_STATE 生成

一种推导相应公式的方法是使用卡诺图 (表 2-5 和 表 2-6)。生成的公式由 OR 运算符连接，并输入到 CLB 工具中。这些公式不需要简化为最简单的形式。

表 2-5. FRAME\_STATE FSM\_1 卡诺图，状态 s0

		当前输入 ( e1、e0 ) RESPONSE_RE、START_OPERATION 或 ENCODER_CLOCK_COMPLETE			
		0.0	0.1	1.1 <sup>(1)</sup>	1.0
		上一状态 s1, s0	0,0 IDLE	0	1 <sup>(2)</sup>
0,1 TRANSMIT_MA	1 <sup>(3)</sup>		1 <sup>(3)</sup>	0	0
1,1 RECEIVE_SL	1 <sup>(4)</sup>		1 <sup>(4)</sup>	1 <sup>(4)</sup>	1 <sup>(4)</sup>
1,0 TRANSMIT_MA RECEIVE_SL	0		1 <sup>(5)</sup>	1 <sup>(5)</sup>	0

- (1) 对应于无效或意外的 e1:e0 组合。系统设计人员决定这种情况下状态机的行为。  
(2) !s1 & !s0 & e0  
(3) !s1 & s0 & !e1  
(4) s1 & s0。C28x 会强制从状态 1,1 转换回 IDLE。  
(5) s1 & !s0 & e0

表 2-6. FRAME\_STATE FSM\_1 卡诺图，状态 s1

		当前输入 ( e1、e0 ) RESPONSE_RE、START_OPERATION 或 ENCODER_CLOCK_COMPLETE			
		0.0	0.1	1.1	1.0
		上一状态 s1, s0	0,0 IDLE	0	0
0,1 TRANSMIT_MA	0		0	1 <sup>(1)</sup>	1 <sup>(1)</sup>
1,1 RECEIVE_SL	1 <sup>(2)</sup>		1 <sup>(2)</sup>	1 <sup>(2)</sup>	1 <sup>(2)</sup>
1,0 TRANSMIT_MA RECEIVE_SL	1 <sup>(3)</sup>		1 <sup>(3)</sup>	1 <sup>(3)</sup>	1 <sup>(3)</sup>

- (1) !s1 & s0 & e1  
(2) s1 & s0。C28x 会强制从状态 1,1 转换回 IDLE。  
(3) s1 & !s0

来自 FSM\_1 的 OUT 信号对应于空闲状态。

检测编码器的响应是该设计的另一个关键组成部分。图 2-13 中所示的 LUT\_2 负责检测编码器的 ACK。如果 FRAME\_STATE 为 WAIT\_FOR\_ACK (0,1) 并且 ENCODER\_RESPONSE\_RE 变为高电平，则表示检测到 ACK。这样会得到公式：i0 & (i1 & !i2)：

- i0 == 1 : ENCODER\_RESPONSE\_RE 变为高电平
- (!i2 & i1) == FRAME\_STATE == WAIT\_FOR\_ACK (0,1)

### 2.3.5.3 CLB\_SPI\_CLOCK 生成

CLB 负责检测编码器的响应，然后为 SPI 提供时钟以接收数据。首先考虑时钟的占空比和频率。图 2-14 展示了生成 SPI 时钟的 CLB 逻辑。图 2-15 展示了相应的仿真波形。

第一步是生成具有指定频率和占空比的时钟（称为 SPI\_CLOCK）。在生成 SPI\_CLOCK 时：

- **TILE3\_COUNTER\_1**：负责时钟的占空比和频率（宽度）。SPI\_CLOCK 高电平转换和低电平转换之间的时间以 CLB 时钟数进行测量。使用以下匹配信号：
  - 零匹配：对应于时钟的低电平转换
  - **match1**：对应于时钟的高电平转换
  - **match2**：对应于时钟的宽度
- **TILE3\_FSM\_1**：根据 **COUNTER\_1** 匹配值生成时钟边沿：**match1** 和零匹配
- **TILE3\_LUT\_0**：达到 **COUNTER\_1** **match2** 输出所指示的时钟宽度时复位计数器

第二步是将时钟与接收到的数据对齐。这是 SPI 正确接收编码器响应所必需的。对齐时钟面临两个挑战：

1. 由于电缆传播延迟，响应可能随时到达，如 节 2.3.1.1 中所述。
2. 编码器还可能延迟确认 **ACK** 以请求额外的处理时间，如 节 2.3.1.2 中所述。

为了对齐时钟，**LUT\_0** 监控上升沿的响应信号。第一个上升沿对应于确认 (**ACK**) 位之后的开始位。检测到此边沿时，**LUT\_0** 会复位 **COUNTER\_1**，从而将 **SPI\_CLOCK** 与响应对齐。

第三步是根据内部 **SPI\_CLOCK** 生成 **CLB\_SPI\_CLOCK**。如 图 2-15 所示。

- **SPI\_CLOCK** 的生成持续时间比 SPI 接收响应所需的时间更长
- **CLB\_SPI\_CLOCK** 仅输出接收响应所需的时钟数

这是通过使用 **TILE4\_FSM\_2** 生成 **SPI\_CLOCK\_OUTPUT\_ENABLE** 信号来实现的。**OUTLUT** 使用此信号来根据需要允许/阻止 **SPI\_CLOCK**。

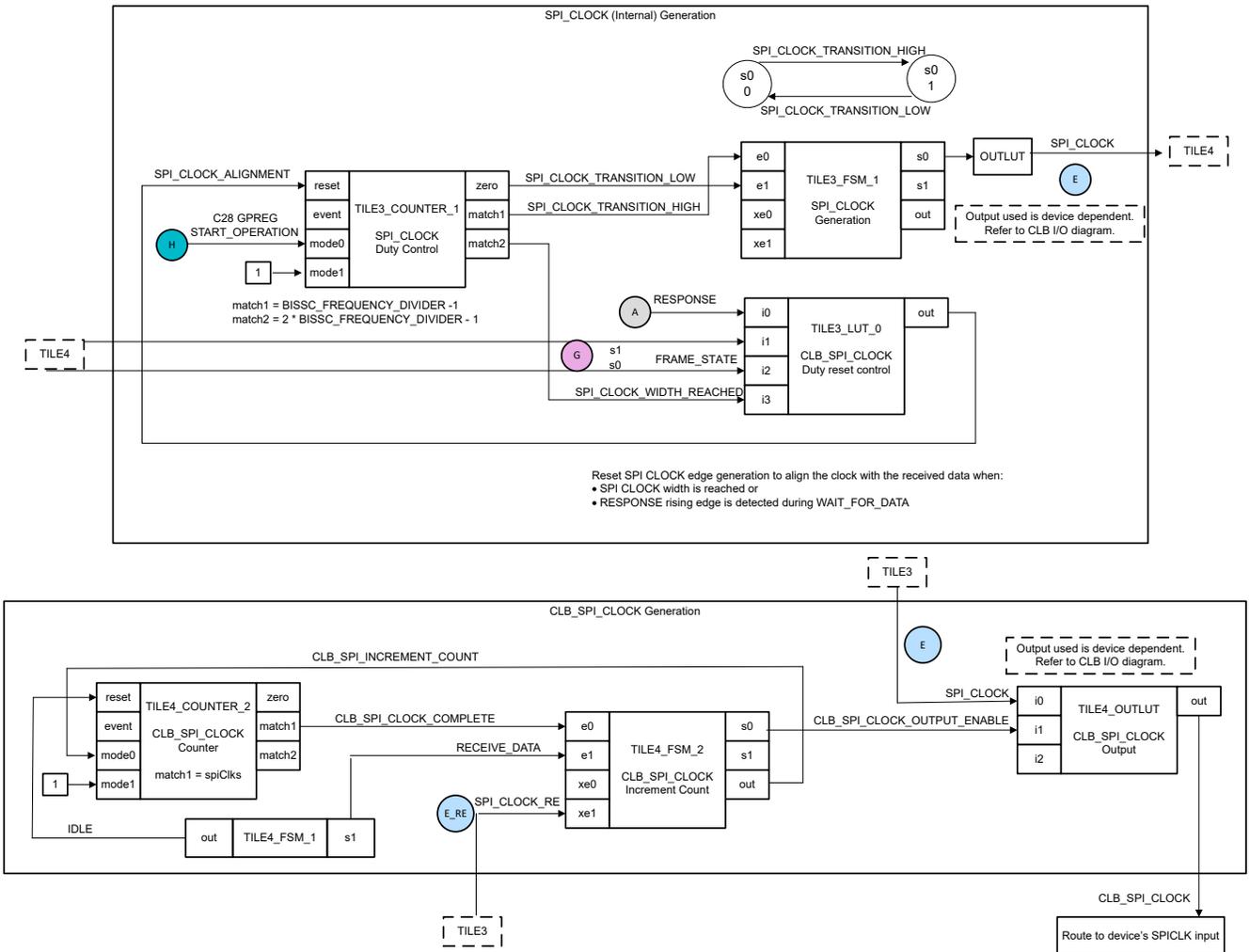


图 2-14. CLB\_SPI\_CLOCK 生成

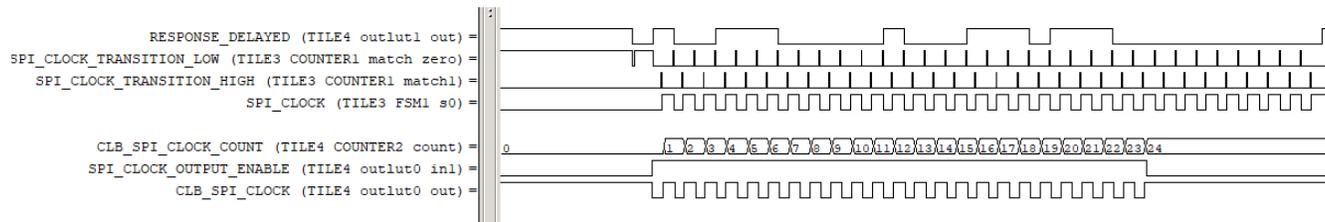


图 2-15. SPI 时钟生成仿真波形

### 2.3.5.4 ENCODER\_CLOCK (MA) 生成

ENCODER\_CLOCK 或 BiSS MA 生成类似于上一节中显示的 CLB\_SPI\_CLOCK 生成。有两个主要差异：

- 生成的 MA 时钟总数取决于接收到响应的时间。
- MA 信号的末端必须保持高电平或低电平，以指示 CDM 位。

图 2-18 中的标记 (1) 指示通过图 2-17 中所示的逻辑检测到编码器的 ACK 的点。在标记 (1) 之前，生成了未知数量的 MA 时钟。从标记 (1) 开始，所需的额外 MA 时钟数量为  $X + 4 + 6$ ，其中：

- X 位：位置位总数 (单圈 + 多圈)
- 4 位：开始位 + CDM 位 + 错误位 + 警告位
- 6 位：CRC 位

该数字由应用程序存储在 TILE4 HLC 寄存器 R0 中。HLC 使用此值以及当前计数器值来调整时钟总数 (COUNTER\_1 match1)。例如，请参阅图 2-18)：

1. 如果  $X = 13$ ，则应用程序在 HLC R0 中存储 23
2. 在标记 (1) 处检测到 ACK 时，将触发 HLC 程序。
3. HLC 读取当前 CLB\_ENCODER\_CLOCK\_COUNT (4)，并添加  $R0 (23) = 27$
4. HLC 将 27 加载到 TILE4 COUNT\_1 match1 中
5. 当 CLB\_ENCODER\_CLOCK\_COUNT 达到 match1 时，MA 时钟的生成完成。

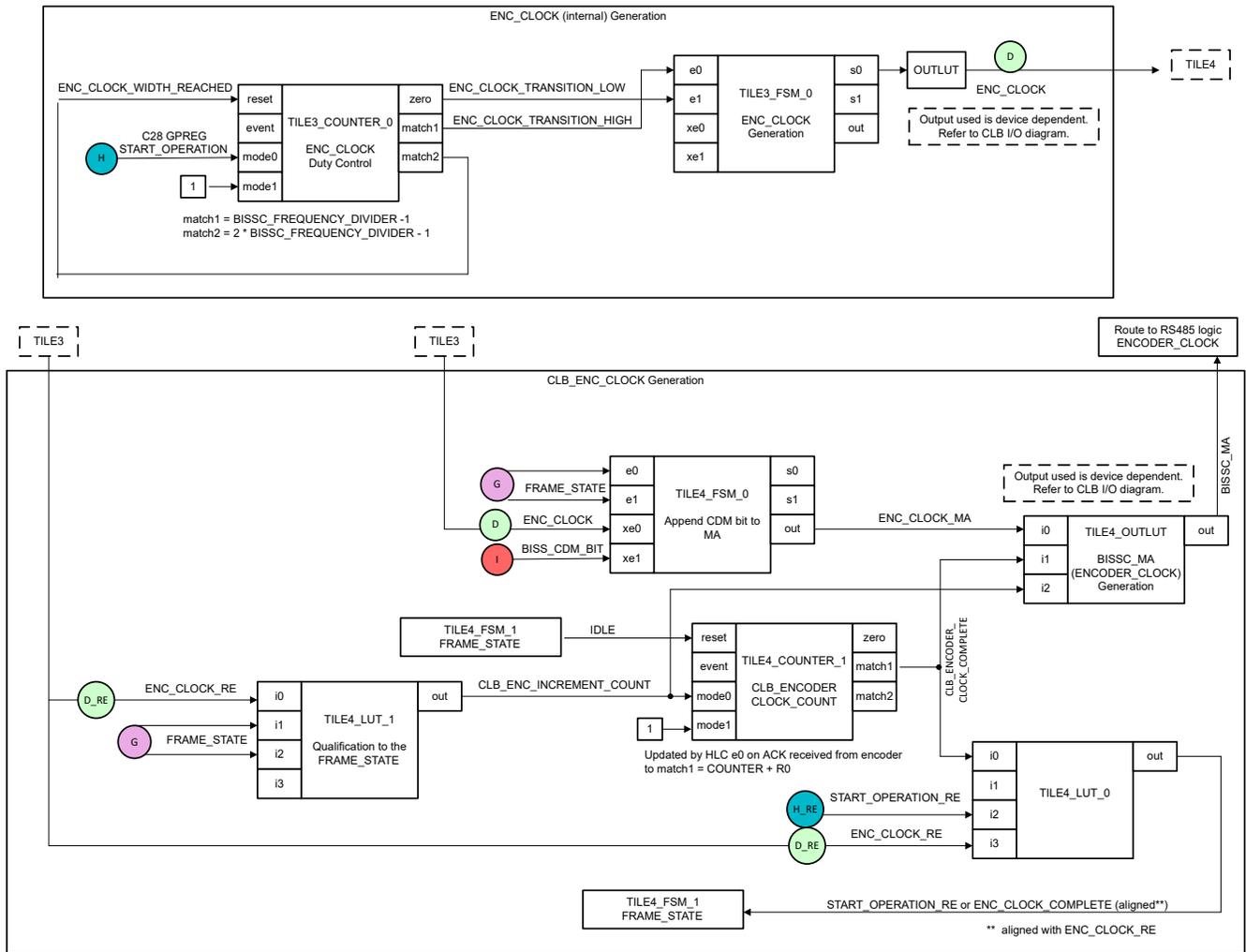


图 2-16. ENCODER\_CLOCK (MA) 生成

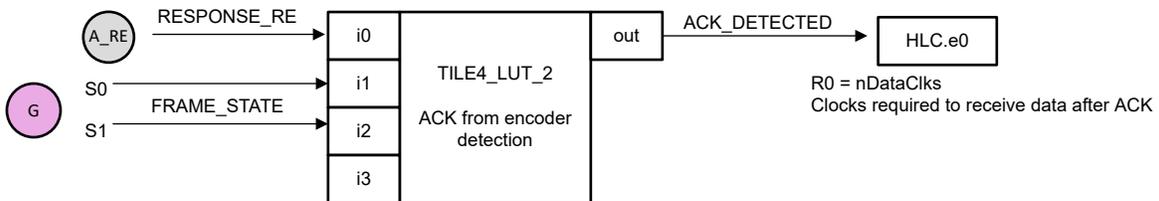


图 2-17. 编码器 ACK 检测逻辑

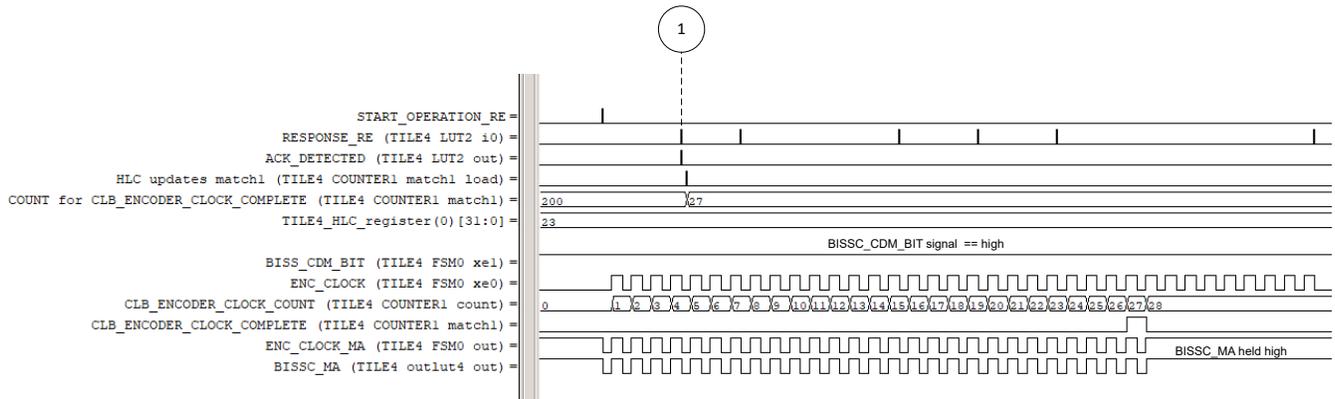


图 2-18. MA 生成仿真波形：CDM 信号高电平

CDM 位是每个 BiSS-C 帧传输到编码器的一位控制数据。MA 信号的末端必须保持高电平或低电平，以指示当前 CDM 位。图 2-18 展示了 BISSC\_CDM\_BIT 电平为高电平时的仿真波形。图 2-19 展示了 BISSC\_CDM\_BIT 信号为低电平时的仿真波形。BISS\_CDM\_BIT 信号由写入逻辑块的 GPREG 的 C28x CPU 控制。

**备注**

对于该讨论，信号状态（高电平或低电平）是指 BISS\_CDM\_BIT 信号的电平。编码器将实际 CDM 位值解释为信号电平的反转。例如，如果 BISS\_CDM\_BIT 信号为高电平，则编码器会检测到 CDM 位值为 0。软件库考虑了这种反转。

当 FRAME\_STATE 因 CLB\_ENCODER\_CLOCK\_COMPLETE (图 2-16) 而改变时，逻辑块 4 FSM\_0 将 BISS\_CDM\_BIT 电平附加到 MA 信号。在 BISSC\_CDM\_BIT 电平为低电平时，输出 (图 2-19) 从 ENC\_CLOCK\_MA 中删除一个额外的边沿。

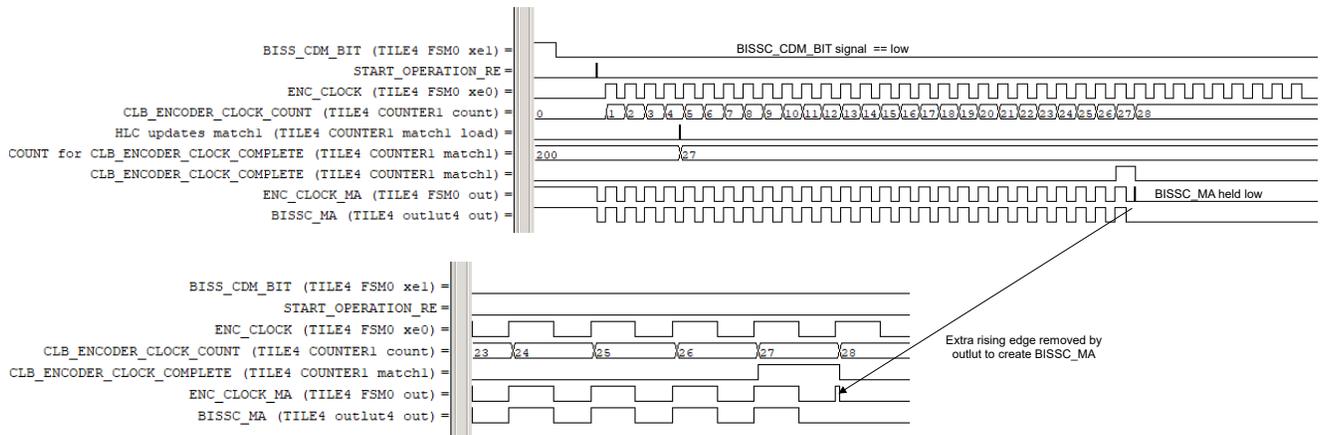


图 2-19. MA 生成仿真波形：CDM 信号低电平

**2.3.6 PM BiSS-C 接口库**

PM BiSS-C 编码器接口库提供以下内容：

- CLB 逻辑实现。
- 定义明确的应用程序编程接口 (API)，使 C2000 器件能够与 BiSS-C 位置编码器通信。

本节提供了 API 中函数的高级概述。

### 2.3.6.1 PM BiSS-C 库函数

BiSS-C 库包含以下函数，用户可以使用这些函数与 BiSS-C 编码器交互。有关 API 的详细说明，请参阅：*C2000 BiSS-C 编码器接口软件指南* ([HTML](#)、[PDF](#)) 表 2-7 简要概述了 API。

**表 2-7. BiSS-C 库函数**

名称	说明
<b>初始化函数</b>	
生成 CRC 表	为具有指定位数的给定 CRC 多项式生成查询表。使用了两个表。一个用于单周期数据 CRC 计算，另一个用于命令数据 CRC 计算。
设置外设	配置 CLB 逻辑实现的 BiSS-C 子系统。
初始化参数	初始化从单周期数据中提取位置和 CRC 所需的参数。
<b>运行时函数</b>	
启动操作	通过开始 MA 信号来发起 BiSS-C 传输。
处理命令数据	命令数据状态机处理每个 BiSS-C 帧的 CDM 输出和 CDS 输入。
接收位置	从 SCD 中提取位置、错误和警告以及 CRC。计算 CRC 并将其与接收到的 CRC 进行比较。
设置 SCD 事务	设置 SPI 和 CLB 以进行新的单周期数据传输。

### 3 硬件、软件、测试要求和测试结果

本节详细介绍了测试程序和结果。包括：

- 硬件要求
- 软件要求
- 测试结果
- 基准测试
- 故障排除指南

#### 3.1 硬件

要对 TIDM-1010 进行实验，需要使用以下硬件组件：

- TIDM-1010 BoosterPack ( 也称为 BOOSTXL-POSMGR )
- 外部 5V 直流电源 ( 请参阅表 1-1 )
- F28P65x LaunchPad 开发套件 (LAUNCHXL-F28P65X) 和 USB 电缆
- BiSS-C 编码器
- 用于连接编码器和 TIDM-1010 的电缆
- 定制适配器，用于将母端接电缆连接到引线适配器
- 安装了 CCS 的计算机

### 3.1.1 TIDM-1010 跳线配置

图 3-1 展示了 TIDM-1010/BOOSTXL-POSMGR 板的跳线配置。

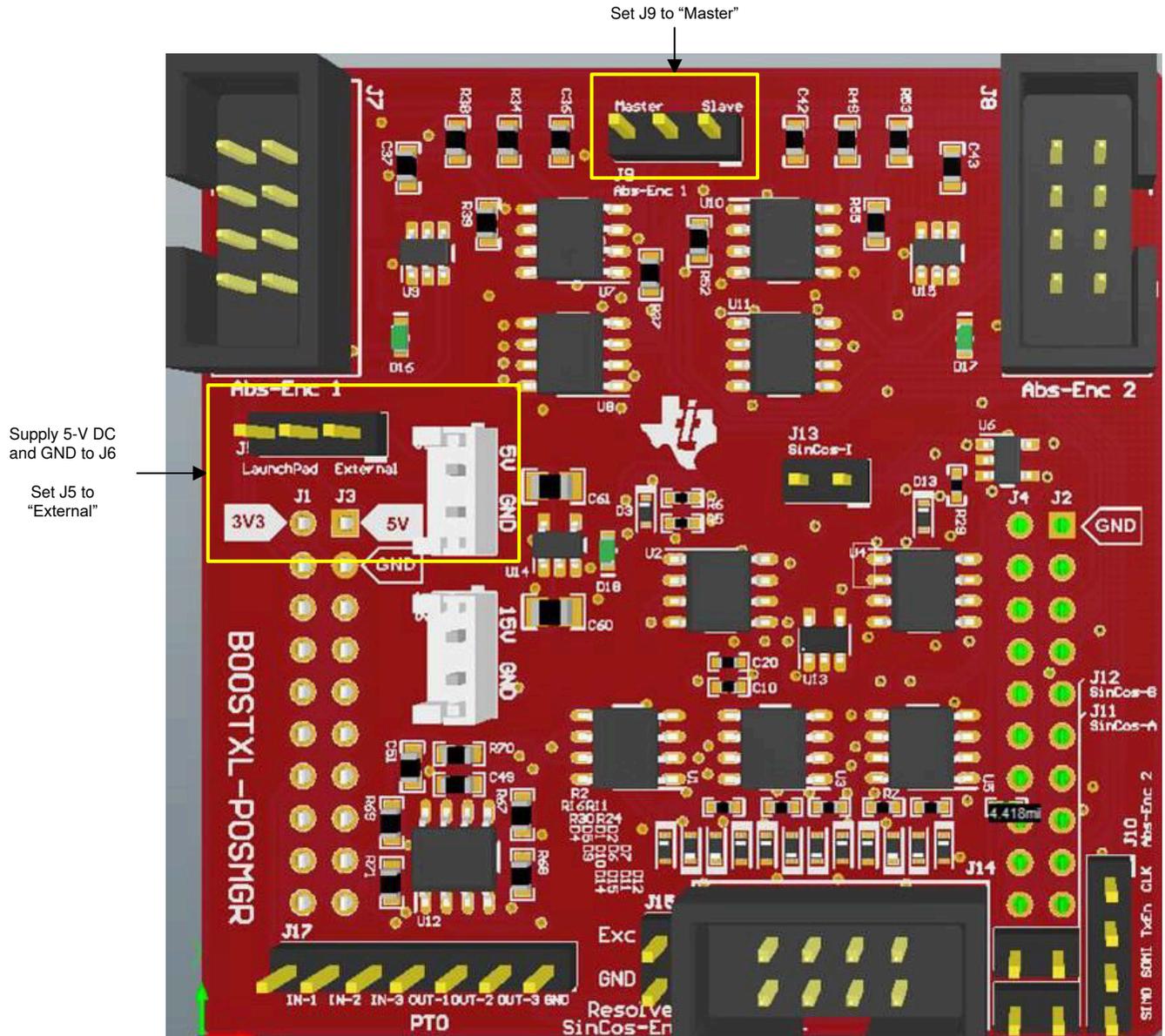


图 3-1. TIDM-1010 板跳线配置

表 3-1 列出了 TIDM-1010 板的跳线配置。

**表 3-1. TIDM-1010 板跳线详细信息**

跳线	功能	位置
J5	TIDM-1010, 5V, 电源平面源选择	外部 <sup>(1)</sup>
J9	Abs-Enc-1 主/从模式选择	主器件 <sup>(2)</sup>
J11	正弦/余弦编码器 A 信号使能	开路
J12	正弦/余弦编码器 B 信号使能	开路
J13	正弦/余弦编码器索引信号使能	开路

(1) 此配置需要用户为 J6 提供外部电源，如 图 3-1 中所示。

(2) 请勿使用从模式选项。此模式的 BoosterPack 逻辑中存在错误。

## 3.2 软件

本节概述了 TIDM-1010 使用的软件。有关完整的文档，请参阅 *C2000 BiSS-C 编码器接口软件指南* ([HTML](#)、[PDF](#))。

该软件指南包括：

- 系统演示代码的文档
- BiSS-C 应用程序编程接口 (API) 的文档
- 将库整合到您自己的解决方案中
- 在双 CPU 器件上将解决方案从 C28x CPU1 移植到 CPU2 的指南。
- 软件更改历史记录

图 3-2 展示了本参考设计中实现的软件架构。采用模块化与可移植方式来实现软件。主要元件包括 C2000 驱动程序库、BiSS-C 编码器接口库、SysConfig GUI 器件配置工具和 CLB 工具。

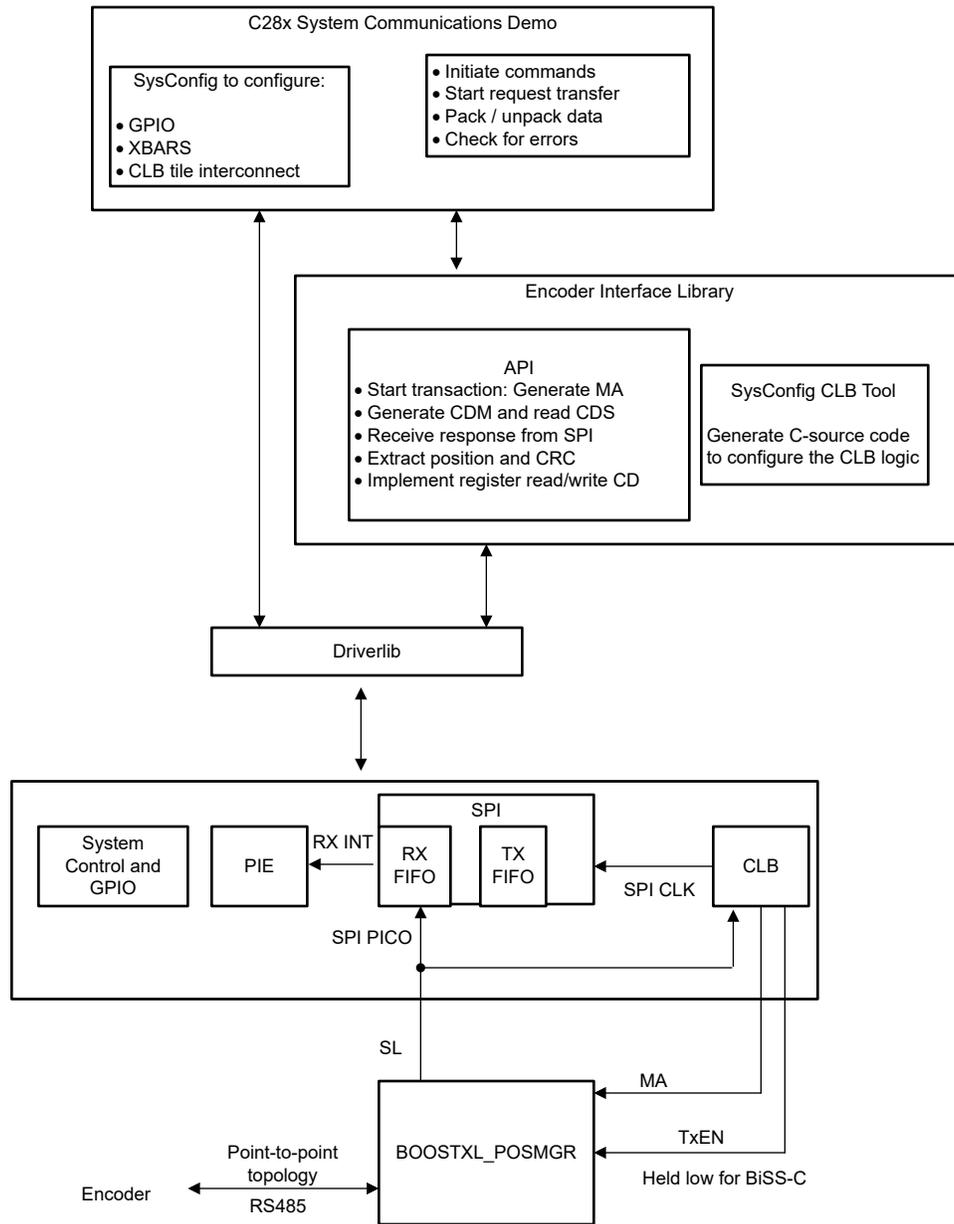


图 3-2. 软件架构

图 3-3 展示了 C2000 BiSS-C 通信演示的流程图。示例应用会配置 C2000 器件，创建控制数据帧，发送 MA 信号，解包响应并检查 CRC。

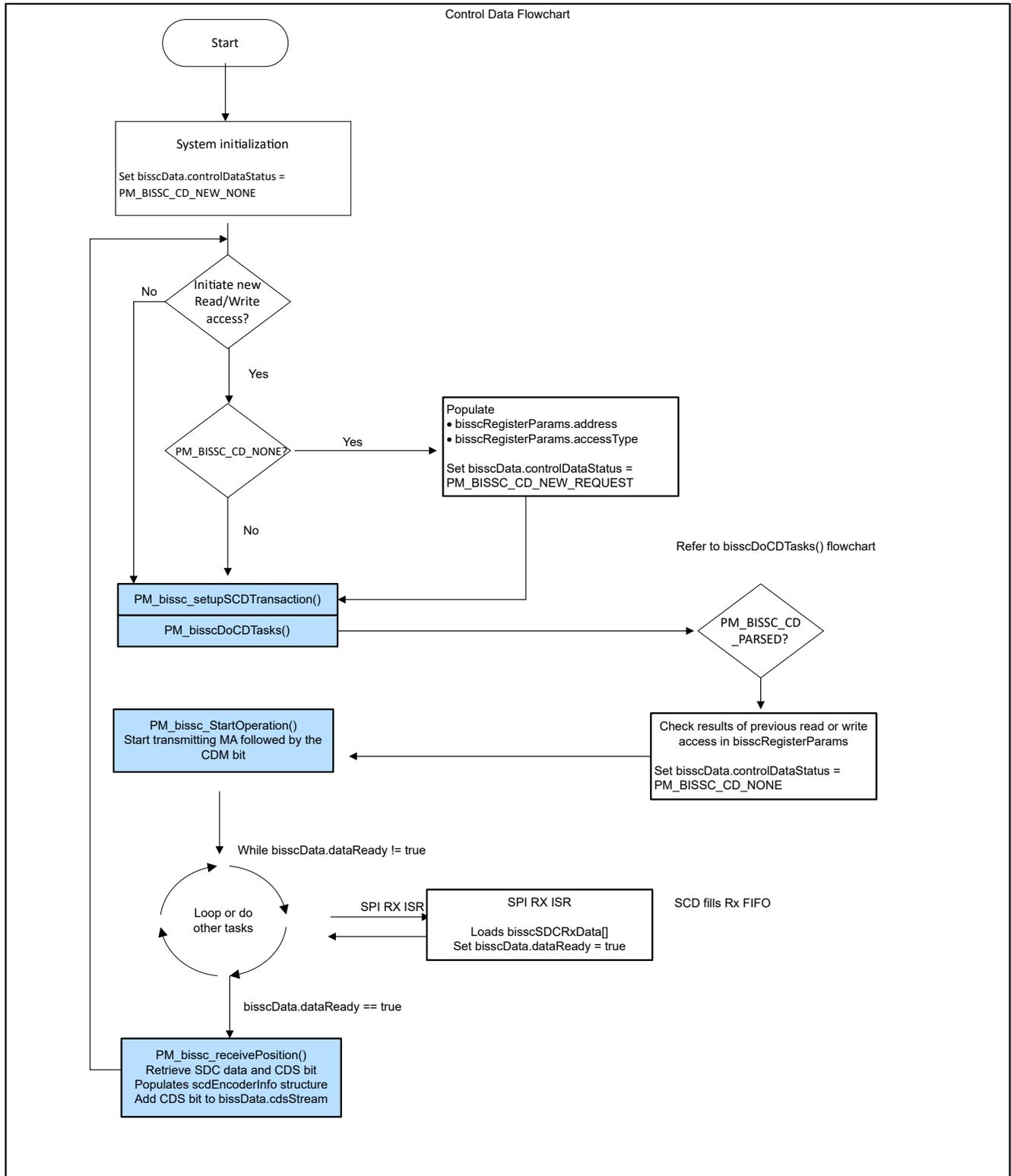


图 3-3. 通信演示流程图

### 3.2.1 C2000 驱动程序库 (DriverLib)

C2000 驱动程序库 (Driverlib) 是一组适用于 C2000 器件系列的低级 API。Driverlib 提供了易于使用的函数调用来配置存储器映射的外设寄存器。C2000Ware 和 C2000Ware 电机控制 SDK 中提供了 Driverlib 的完整源代码。有关更多信息，请参阅 C2000 软件指南的“DriverLib”部分。

### 3.2.2 C2000 SysConfig

C2000 SysConfig 是用于配置 C2000 实时控制 MCU 的图形用户界面工具。SysConfig 会自动生成与 Driverlib 连接的嵌入式软件。在此参考设计中，SysConfig 工具用于生成代码来配置 SPI、GPIO、INPUTXBAR/OUTPUTXBAR 和 CLB MUX。有关更多信息，请参阅 C28x Academy : SysConfig 模块。

### 3.2.3 C2000 可配置逻辑块工具

C2000 CLB 工具支持通过图形界面配置 CLB 逻辑。CLB 工具是一款易于使用的 GUI，内置于 Code Composer Studio 中，并可利用 C2000 SysConfig 插件。此参考设计中使用 CLB 工具为 BiSS-C 编码器接口配置逻辑块，如设计说明中所述。有关更多信息，请参阅 C28x Academy : 可配置逻辑块模块。

### 3.2.4 安装 Code Composer Studio™ 和 C2000WARE-MOTORCONTROL-SDK

1. 安装 CCSTUDIO IDE v12.7.1 或更高版本 ( 如果计算机上还没有 CCSTUDIO ) 。
2. 安装 C2000WARE-MOTORCONTROL-SDK v5.03.00.00 或更高版本 ( 如果尚未在计算机上安装该软件 ) 。
3. 安装后，请参阅 C2000 BiSS-C 编码器接口软件指南 ( HTML、PDF ) 以获取进一步说明。

#### 备注

CLB 工具包含在 Code Composer Studio (sysconfig) 和 SDK 的 C2000Ware 子组件 ( 支持实用程序 ) 中。若要运行基于 CLB 的仿真，需要安装其他工具 ( 请参阅 CLB 工具用户指南中关于这些工具的说明 ) 。

### 3.2.5 查找参考软件

此参考设计中包含的软件由两部分组成：

- 说明编码器接口用法的系统示例。示例项目源文件的位置如表 3-2 所示。
- 编码器接口库。表 3-3 展示了库源文件的位置。

有关完整的文档，请参阅 C2000 BiSS-C 编码器接口软件指南 ( HTML、PDF ) 。

表 3-2. 系统示例的位置

C:\ti\c2000\c2000ware_MotorControl_SDK_[version]\	SDK 的默认安装位置。([SDK])
[SDK]\solutions\boostxl_posmgr\	特定于器件的基础安装目录 ([pm_base])
[pm_base]\[device]\ccs\bissc	Code Composer Studio (CCS) projectspec 文件。用于将工程导入您的 CCS 工作区。

表 3-3. 编码器接口库的位置

C:\ti\c2000\c2000ware_MotorControl_SDK_[version]	SDK 的默认安装位置。([SDK])
[SDK]\libraries\position_sensing\bissc	库基础安装目录 ([lib_base])
[lib_base]\ccs\[device]	参考库的 Code Composer projectspec 文件。使用这些工程可为每个器件重新构建库。

### 3.3 测试和结果

本节详细介绍了测试程序、结果和基准测试。另外还提供了故障排除指南。

#### 3.3.1 硬件配置

1. 确保 TIDM-1010 器件的跳线配置如 [表 3-1](#) 中所述。
2. 使用 BoosterPack 站点 2 连接器 ( J1、J3 和 J4、J2 ) 将 TIDM-1010 器件连接到 LaunchPad。确保将 TIDM-1010 器件连接到 LaunchPad 的站点 2，如 [图 3-4](#) 中所示。

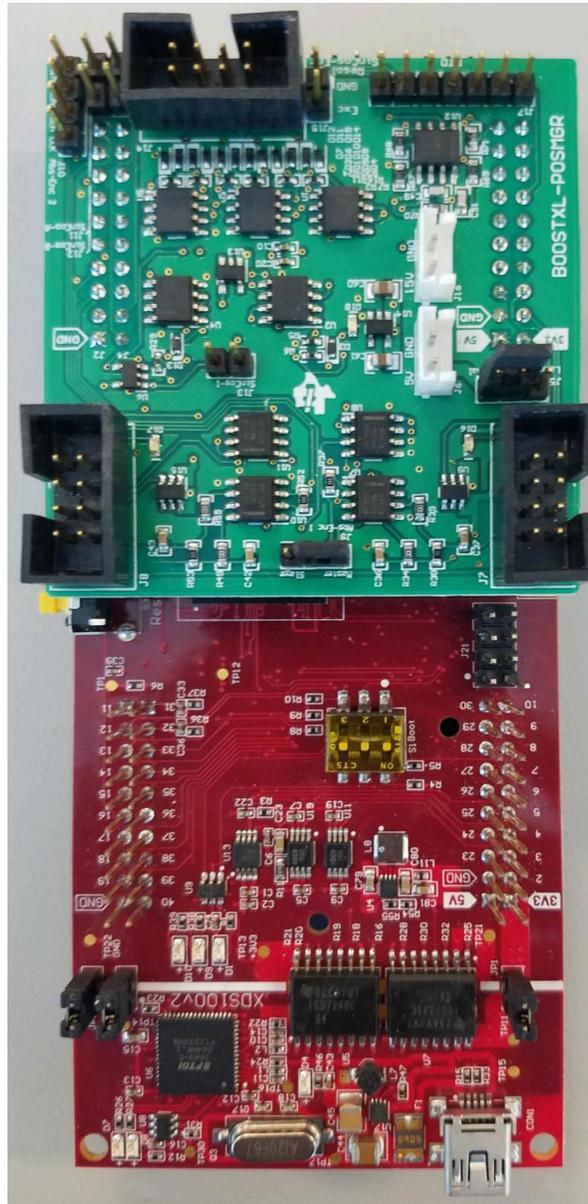


图 3-4. 连接到 LaunchPad™ 的站点 2 的 TIDM-1010 板

3. 将 USB 电缆连接到 LaunchPad。
4. 连接到编码器：
  - a. 准备一个适配器，使用圆形母型转引线适配器将电缆连接到 BiSS-C 接口 ( 有关用于编码器连接器 J7 的接头，请参阅 BOM )。
  - b. 将 CLK+/CLK- 连接到编码器的 BiSS-C MA+/MA- 信号。
  - c. 将 DATA+/DATA- 连接到编码器的响应 SL+/SL-。
  - d. 插入在上一步中创建的适配器的接头，以连接到 Abs-Enc-1 (J7)。电缆的母端连接到编码器。[图 3-5](#) 展示了 J7 的引脚排列。

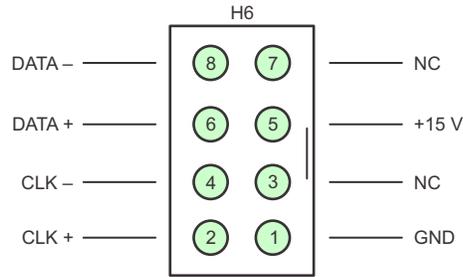


图 3-5. TIDM-1010 板上的 Abs-Enc-1 (J7) 引脚排列

- 5V 直流电源和 GND 连接到 J6，如图 3-1 中所示。板现在看起来像图 3-6。BoosterPack LED D18 亮起，表明板已通电。

**备注**

对于某些编码器，BoosterPack 可能无法在上电时提供足够的电流。如果编码器未做出响应，请尝试将 BoosterPack 外部的电源连接到编码器。如果这样做，请将公共接地连接到 BoosterPack。

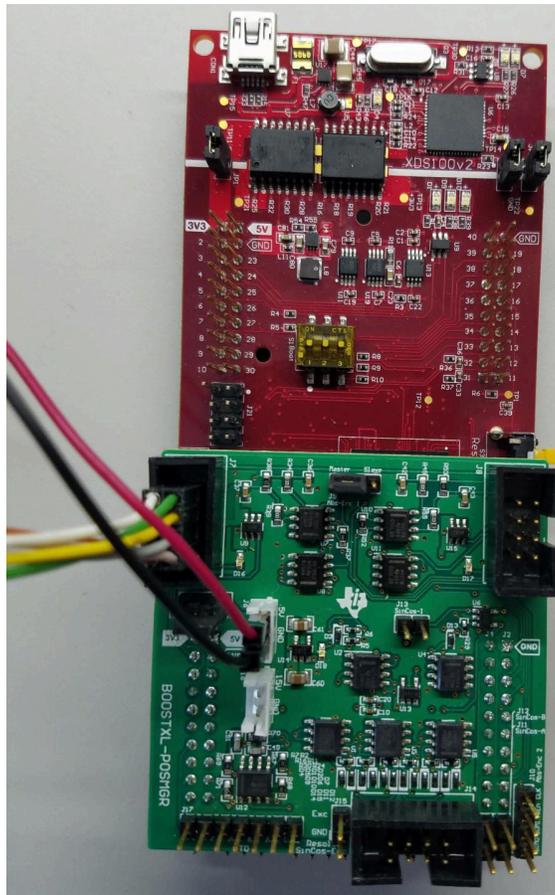


图 3-6. 连接到 BiSS-C 编码器的 TIDM-1010 板

### 3.3.2 构建和加载工程

按照 C2000 BiSS-C 编码器接口软件指南 ( [HTML](#)、[PDF](#) ) 中的说明来加载和运行系统解决方案。请参阅软件指南的系统解决方案部分。

这些说明包括：

- 将项目导入到适用于器件的 Code Composer Studio (CCS) 中
- 配置库和系统示例
- 选择构建配置
- 填充观察 (表达式) 窗口
- 运行代码

### 3.3.3 运行示例代码

BiSS-C 系统解决方案是仅用于通信的演示。该演示会向编码器发送 MA 信号，接收响应并检查错误。该模式在 `while(1){}` 循环中重复。此外，该演示会定期发送对编码器寄存器的读取/写入访问。根据提供的信息，使用组选择寄存器。在运行演示时，您可以在手动转动电机或编码器轴的同时使用逻辑分析仪或示波器 ( [图 3-7](#) ) 来监控 MCU 的输出信号。

#### 备注

仅 F2837xD 需要在 CLB 和 SPI 时钟之间进行外部连接。其他器件在 CLB 和 SPICLK 之间具有内部连接。对于具有内部连接的器件，SPICLK 也可引出至一个引脚进行监控。[节 2.3.4.1](#) 中的器件输入/输出图展示了 SPI CLK 的测试连接。

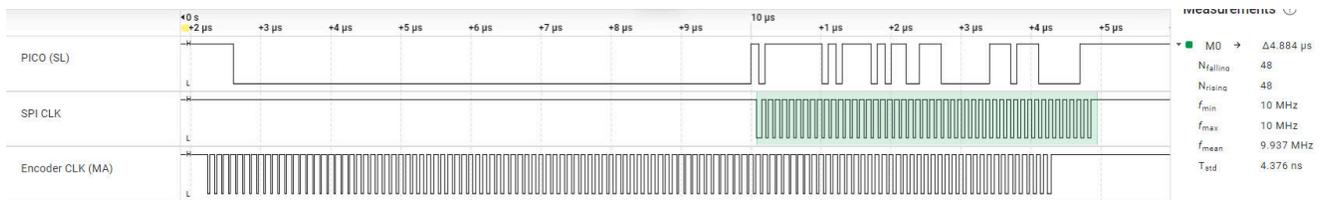


图 3-7. BiSS-C 波形

图 3-7 中所示的波形为：

- 对于具有 32 个位置位的编码器
- 具有 10MHz MA 信号的 10m 电缆
- SPI 时钟引出至引脚进行监控
- SPI FIFO 配置为 12 位字，并在 4 级满时产生中断
- 将中断置为有效需要 48 个时钟
- 由逻辑分析仪以 100MHz 的采样频率捕获

### 3.3.4 编码器测试

[表 3-4](#) 列出了使用各种类型编码器进行的测试。这些测试是在德州仪器 (TI) 执行的。测试包括读取位置、读取寄存器、写入寄存器。可在电缆长度高达 10m 时进行 10MHz 操作。降低频率后，还测试了更长的电缆长度。

最大 MA 频率取决于编码器和编码器电缆。电缆的质量会影响编码器的数字通信性能和可实现的覆盖范围。例如，编码器电缆中的电源线可能具有更大的尺寸，以更大限度减小长电缆的压降。通常，编码器供应商会为其编码器推荐相应的电缆。电缆长度不低于 100m 时，这种电缆质量对于工作至关重要。

有关 MA 时钟频率与电缆长度之间的关系指南，请参阅文档 *BiSS 接口：AN15：BiSS 接口网站提供了 BiSS C 主器件操作详细信息*。

表 3-4. 编码器测试汇总

编码器制造商	编码器名称	类型	分辨率 (位)
Lika	HS58S18	旋转	18 位
Lika	EH036-20-12	旋转	32 位 (12 MT + 20 ST)

**表 3-4. 编码器测试汇总 (续)**

编码器制造商	编码器名称	类型	分辨率 (位)
iCHaus	EVAL MHM1D	评估板	28 位 (16 MT + 12 ST)

### 3.3.5 基准测试

表 3-5 列出了从 RAM 执行 BiSS-C 库函数所需的 C28x CPU 周期。此数据是使用 ERAD 模块和以下编译器设置收集的：

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

#### 备注

这些函数可根据应用进一步优化。例如，如果在编译时而不是运行时知道编码器的分辨率，则可以省去一些代码。

**表 3-5. 周期数基准测试**

功能	周期 -O2 -mf2	注释
PM_bissc_setupSCDTransaction	369	
bissc_setupCDTest	23	
PM_bissc_receivePosition	552	取决于编码器的分辨率。此测量使用了具有 12 个多圈位和 20 个单圈位的编码器。
PM_bissc_doCDTasks	39 (最小值)、225 (最大值)	寄存器读取或寄存器写入完成时，会出现最大周期。
PM_bissc_startOperation	39	
PM_bissc_setFreq	73	
bissc_getCRC	90	
bissc_spiRxISR	497	

表 3-6 列出了与每个库源文件相对应的 16 位字代码大小。C28x 查询表占用 256 个字的 RAM 或闪存，此表中并未反映。

**表 3-6. 16 位字的代码大小**

源文件	代码大小 -O2 -mf2
pm_bissc_source	910
pm_bissc_crc	49
clb_config	463

### 3.3.6 故障排除

检查以下波形有助于进行故障排除。请参考设计说明中的 I/O 图：

- CLB 生成的 SPI 时钟。这是在测试引脚上引出的，用于观察。
- SPI 输入引脚上的编码器响应。
- RS485 线路驱动器和编码器之间的编码器响应 (SL+/SL-)。注意：数据是差分信号。因此，需要使用特殊的探头来观察。
- TxEN 信号。确认此信号保持低电平。BiSS-C 设计不会将 TxEN 拉高。
- RS485 线路驱动器和编码器之间的编码器时钟 (MA+/MA-)。注意：数据是差分信号。因此，需要使用特殊的探头来观察。

1. 如果未传输 MA 信号：

- 确定问题出在 RS485 线路驱动器之前还是之后。
  - 如果问题出在线路驱动器和编码器之间，则确认 TxEN 保持低电平。
2. 如果 MA 频率不符合预期：
    - 检查 `bissc.h` 中的 `BISSC_MA_CLOCK` 定义和 `BISSC_FREQ_DIVIDER` 定义
  3. 如果未看到编码器响应：
    - 检查编码器的电源连接。
    - 根据编码器规格检查电源电流。
    - 降低 MA 的频率并重试。这可能表明电缆或连接器存在问题。
    - 确认电缆设计和长度满足编码器制造商的要求。
    - 确定问题出在 RS485 线路驱动器之前还是之后。
  4. 如果观察到响应波形，但 SPI 未捕获到该波形：
    - 如果 CLB 设计移到不同的逻辑块或移植到不同的器件，则可能发生这种情况。
      - 检查来自 CLB 的 SPI 时钟。当 CLB 看到编码器的响应时，该时钟开始。确认内部 XBAR 连接正确地将响应路由到 SPI 和适宜 CLB 输入。
      - 如果看到 SPI 时钟，请确认 CLB 输出的内部连接到所使用的 SPI 实例。这会根据所使用的 CLB 逻辑块而改变。

## 4 设计文件

要下载设计文件，请参阅位于 [TIDM-1010](#) 的产品页面。

## 5 软件文件

软件源文件包含在适用于 [C2000™ MCU](#) 的 [MotorControl 软件开发套件 \(SDK\)](#) 中。

## 6 相关文档

1. iCHaus。 [iC-Haus](#)，网站。
2. BiSS 接口 [BiSS 接口](#)，网站。
3. C2000 BiSS-C 编码器接口软件指南 ( [HTML](#)、[PDF](#) )
4. Wikipedia， [循环冗余校验](#)，页面。
5. 德州仪器 (TI)： [Code Composer Studio](#) 页面
6. 德州仪器 (TI)： [C28x Academy](#) 可提供易于使用的培训模块，涵盖有关所有 C28x 器件的各类主题。
7. 德州仪器 (TI)： [C2000 软件指南](#) 包括 C2000 软件、软件开发套件和开发工具的概述。
8. 德州仪器 (TI)：适用于工业驱动和电机控制的 [C2000 DesignDRIVE](#) 软件
9. 德州仪器 (TI)： [C2000 Position Manager SinCos 库](#) 用户指南

## 商标

C2000™, BoosterPack™, LaunchPad™, and E2E™ are trademarks of Texas Instruments.  
BiSS-C™ is a trademark of iC-Haus GmbH.

所有商标均为其各自所有者的财产。

## 7 术语

<b>C28x</b>	指具有 C28x CPU 内核的器件。
<b>API</b>	应用程序编程接口。启用编码器接口协议的库的定义。
<b>BiSS</b>	传感器和执行器的开源数字接口。在本文档中，BiSS 接口或 BiSS 是指 2007 年推出的 BiSS-C 协议。BiSS-C 表示双向串行同步连续模式。有关协议和实现方面的更多详细信息，请参阅 <a href="#">BiSS 接口</a> 网站。
<b>BiSS 帧</b>	单周期数据加上一位控制数据。BiSS 帧在每个 BiSS 周期作为响应发送。
<b>BiSS 许可证</b>	购买 TI MCU 时不会自动包含 BiSS IP 使用许可证。BiSS 接口的用户必须从 <a href="#">BiSS 接口</a> 申请自己的许可证。
<b>CD</b>	来自 BiSS-C 规格。控制数据。在每个 BiSS 帧传输一位的数据。例如，CD 允许读取和写入编码器寄存器。
<b>CDM</b>	来自 BiSS-C 规格。主器件控制数据。一位控制数据 CD，在每个 BiSS 帧通过控制器发送到编码器。
<b>CDS</b>	来自 BiSS-C 规格。从器件控制数据。一位控制数据 CD，在每个 BiSS 帧通过编码器发送回控制器。
<b>CLB</b>	C2000 器件上的可配置逻辑块外设。
<b>CRC</b>	循环冗余校验
<b>CTS</b>	来自 BiSS-C 规格。控制选择位。BiSS-C 控制数据 (CD) 帧，以一个开始位开始，后跟 CTS 位。如果 CTS 为 0，则控制帧为命令帧。如果 CTS 为 1，则控制帧为寄存器访问帧。
<b>MA</b>	来自 BiSS-C 规格。主时钟。该脉冲序列输出从控制器发送到编码器/传感器。它由编码器时钟以及来自主器件的一位控制数据组成。
<b>Position Manager BoosterPack</b>	用于连接各种位置编码器的未来 EVM。TIDM-1010 板与 Position Manager BoosterPack EVM 相同 ( 请参阅 <a href="#">节 2.3.3</a> )
<b>PM</b>	Position Manager - C2000 器件上用于位置编码器接口的基础硬件和软件

<b>PM_bissc 或 PM_BISSC</b>	用于所有库函数的前缀。
<b>点对点拓扑</b>	仅一个器件 ( 编码器或传感器 ) 与主器件相连的配置。
<b>SSI</b>	串行同步接口
<b>SL</b>	来自 BiSS-C 规格。从器件返回。从器件向 BiSS-C 主器件输入的数据。这是编码器响应，包括位置和每个 BiSS 帧的一位 CD 数据。
<b>SPI</b>	C2000 器件上的串行外设接口外设。

## 8 关于作者

**LORI HEUSTESS** 多年来一直是 C2000 团队的一员。她感兴趣的领域包括 CPU 和外设验证、软件开发和工业应用。Lori 目前在应用特定 MCU (ASM) 工业应用团队工作。

**SUBRAHMANYA BHARATHI AKONDY** 从事过几种 C2000 MCU 产品和控制外设的架构定义和设计。擅长 MCU 架构、应用和设计方面。

**SHEENA PATEL** 在 C2000 MCU 组的工业驱动器团队中担任产品营销工程师。

## 9 修订历史记录

Changes from Revision * (April 2018) to Revision A (November 2024)	Page
• 更新了整个文档中的表格、图和交叉参考的编号格式.....	1
• 更新了 LaunchPad 信息.....	1
• 添加了指向 BiSS 接口网站的链接。添加了库高级功能。.....	1
• 添加了系统概述 简介部分.....	4
• 更新了方框图。.....	4
• 添加了主要产品 简介部分.....	4
• 在主要产品 部分中添加了指向器件的超链接.....	4
• 将 MCU 和 LaunchPad 信息更改为 F28P65x.....	4
• 添加了设计注意事项 简介部分.....	5
• 添加了 BiSS-C 协议 部分.....	5
• 添加了线路延迟补偿 部分.....	6
• 添加了编码器处理时间请求 部分.....	7
• 添加了控制通信 部分.....	7
• 添加了 C2000 BiSS-C 编码器接口概述 部分.....	8
• 添加了功能概述。更新了表中的第 3 列 TIDM-1010 板和 BOOSTXL-POSMGR 连接器，指示 TIDM-1010 的专门用法。添加了 BoosterPack 方框图.....	10
• 添加了 MCU 资源要求 部分.....	12
• 添加了输入、输出信号和 CLB 逻辑块 部分.....	12
• 添加了 CLB BiSS-C 实现详细信息 部分.....	14
• 添加了事务波形 部分.....	14
• 添加了 FRAME_STATE 生成 部分.....	15
• 添加了 CLB_SPI_CLOCK 生成 部分.....	18
• 添加了 ENCODER_CLOCK (MA) 生成 部分.....	20
• 添加了 PM BiSS-C 接口库 简介部分.....	22
• 删除了 C2000 BiSS-C 编码器接口软件指南 中的详细信息。添加了软件指南超链接。添加了库函数汇总。...	23
• 添加了硬件 简介部分.....	24
• 更新了 TIDM-1010 板跳线详细信息表 (注释 2，删除了未使用的注释 3).....	25
• 添加了 C2000 BiSS-C 编码器接口软件指南 的摘要和超链接。添加了软件架构和通信演示流程图。.....	26
• 添加了 C2000 驱动程序库 (DriverLib) 部分.....	29
• 添加了 C2000 SysConfig 部分.....	29
• 添加了 C2000 可配置逻辑块工具 部分.....	29
• 将安装 CCS 和 controlSUITE 部分更改为安装 Code Composer Studio™ 和 C2000WARE-MOTORCONTROL-SDK 部分.....	29
• 添加了查找参考软件 部分.....	29
• 删除了 C2000 BiSS-C 编码器接口软件指南 中的说明。添加了软件指南概述。.....	32
• 更改了 C2000 BiSS-C 编码器接口软件指南 中的说明以及示例应用摘要。.....	32
• 更新了编码器测试表数据。更新了电缆长度建议。.....	32
• 添加了基准测试 部分.....	33
• 添加了故障排除 部分.....	33
• 更新了超链接.....	35
• 添加了指向 iCHaus、BiSS 接口、Code Composer Studio、C28x Academy 和 C2000 软件指南的超链接。更新了指向 BiSS-C 软件指南的超链接。.....	35
• 添加了对 API、BiSS 帧、BiSS 许可证、CD、CDM、CDS、CTS、MA、SLA、SL 等的更多定义.....	35
• 已将 Lori Heustess 添加到作者列表.....	36

## 重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2024，德州仪器 (TI) 公司