

# NCN26010 - 入门

## 基本配置、通信和异常处理

### AND90155/D

#### 导语

NCN26010 10Base-T1S 器件符合 IEEE 802.3cg 规范以及 OPEN 联盟技术委员会6 (Open Alliance, TC6) 的 SPI (串行外设接口) 协议, 并具有各种可选附加功能。

为了正确可靠地运行, 在配置 NCN26010 时有一些注意事项, 特别是因为该器件在没有正确配置的情况下无法参与多点数据段 (multi-drop segment) 通信。此外, 如果配置不当, 还有可能长期干扰整个数据段。

本应用笔记旨在为用户提供针对其特定应用的 NCN26010 配置指南。

这里仅描述了所需的基本设置, 并通过配置示例对这些设置进行了讨论。

#### 参考文献

|     |                                                                                                                              |
|-----|------------------------------------------------------------------------------------------------------------------------------|
| [1] | IEEE802.3cg-2019<br>"IEEE 以太网标准第 5 修正案:<br>通过单对平衡导线实现 10 Mb/s<br>运行及相关供电传输的物理层规范和管理参数"<br>IEEE 计算机协会, ISBN 978-1-5044-6420-8 |
| [2] | OPEN 联盟<br>TC6 - 10BASE-T1x MACPHY 串行接口 1.0 版, 2020<br>年 9 月 14 日发布                                                          |

#### 工作模式

NCN26010 提供 IEEE 802.3cg 标准规定的强制和可选工作模式, 并提供扩展功能:

- CSMA/CD 作为基本工作和后备协议
- 物理层冲突避免机制 (PLCA)
  - ◆ 突发模式
  - ◆ 优先级模式
- 增强噪声抗扰度 (ENI)

作为一款 MAC-PHY (物理层-数据链路层) 器件, NCN26010 在一个单一芯片中集成了 10Base-T1S 物理层器件 (Physical Layer Device) 和符合 IEEE802.3 第 4 款的媒体访问控制器 (Media Access Controller)。这种组合为低成本 MCU 提供了以太网通信功能, 具备至少运行在 15 MHz 且可软件实现 TCP/IP 协议栈 (例如, FreeRTOS+) 的 SPI 接口。

我们将重点介绍 OPEN 联盟 MACPHY SPI 协议的基本功能。有关详细信息, 建议用户查阅的 OPEN 联盟 TC6 文档 1.0 版。

为了说明该部分的基本用途, 本应用笔记首先讨论了如何读写配置寄存器, 以及如何通过 SPI 接口发送实际以太网帧数据。

接下来将介绍一组基本配置以及 MAC 内部地址过滤器 (Address Filter) 和过滤器掩码 (Filter Mask) 的使用。

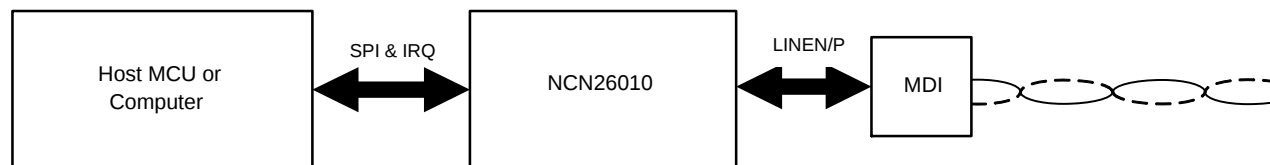


图 1. 简化框图

## 应用信息

### SPI 接口

NCN26010 实现了 OPEN 联盟 TC6 SPI 协议的所有必备功能。

该协议使用 SPI 接口接收和发送以太网帧，并通过所谓的控制命令事务交换配置数据。

通过查看 32 位长通信报头 (Header) 的最高有效位 (MSB)，可以区分控制事务和数据事务。

### 控制命令报头

当第 31 位 (DNC = 数据 - 非命令) 被置为 0 时，表示这是一个控制命令。控制命令的格式如下：

|            |    |     |     |     |    |    |    |     |             |    |    |    |    |    |    |
|------------|----|-----|-----|-----|----|----|----|-----|-------------|----|----|----|----|----|----|
| 31         | 30 | 29  | 28  | 27  | 26 | 25 | 24 | 23  | 22          | 21 | 20 | 19 | 18 | 17 | 16 |
| 0          | 0  | WNR | AID | MMS |    |    |    |     | ADDR [15:8] |    |    |    |    |    |    |
| 15         | 14 | 13  | 12  | 11  | 10 | 9  | 8  | 7   | 6           | 5  | 4  | 3  | 2  | 1  | 0  |
| ADDR [7:0] |    |     |     |     |    |    |    | LEN |             |    |    |    |    |    | P  |

图 2. 控制命令格式

### 数据报头

为了传输以太网数据 (双向)，上位机需要发送有效的数据传输报头，随后紧跟实际的以太网数据。数据

传输报头的特征是第 31 位被置 1。以下是与 NCN26010 器件使用相关的数据报头字段。

|    |    |      |    |    |    |    |    |    |    |    |    |     |    |    |    |
|----|----|------|----|----|----|----|----|----|----|----|----|-----|----|----|----|
| 31 | 30 | 29   | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19  | 18 | 17 | 16 |
| 1  | 0  | NORX | 0  | 0  | 0  | 0  | 0  | 0  | 0  | DV | SV | SWO |    |    |    |
| 15 | 14 | 13   | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3   | 2  | 1  | 0  |
| 0  | EV | EBO  |    |    |    |    |    | 0  | 0  | 0  | 0  | 0   | 0  | 0  | P  |

图 3. 数据报头字段

需要注意的是，在 NCN26010 的应用中，只使用 NORX、DV、SV、SWO、EV、EBO 和强制 (奇数) 校验 P 字段。所有其他位建议始终发送 0。

关于字段名称及其功能的详细说明，请参阅 OPEN 联盟 TC6 文档，rev 1.0 版本。

### 读写配置和状态寄存器

要写入或读取单个寄存器，总共需要向 NCN26010 10Base-T1S MACPHY 发送 12 个字节。

控制命令以命令控制报头开头，随后是：

- 对于寄存器写入控制事务 (寄存器写入)，包含 32 位寄存器内容和 32 位占位数据 (dummy data)
- 对于读取控制事务 (寄存器读取)，包含 64 位占位数据

SPI 作为一种双向全双工通信方式，每传输/接收一位数据就需要一个 SCK (串行时钟) 跳变。为了在寄存器读写事务中接收适当数量的字节，MCU 上位机需要通过 MOSI 线路向 NCN26010 器件发送占位字节 (dummy bytes)。

连接的 MCU 必须能够在不断开 MACPHY 的 SPI 芯片选择的情况下完成整个事务。

为寄存器读写准备两个 12 字节长的数组似乎最为简单。在数组中填入正确的数据 (报头和写入数据)，然后通过 SPI 以 12 字节的批量传输方式发送，同时将结果接收到 12 字节的接收缓冲区中。

## AND90155/D

寄存器读取示例:

应用程序希望读取位于 MMS0 地址 0x0001 的 MACPHY 的 SPI 识别寄存器。

首先, 我们需要生成一个适当的报头: 按照此主题确定报头:

|            |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Field      | 31                                  | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WNR        | 0                                   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| AID        | 0                                   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MMS        | 0                                   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ADDRESS    | 0                                   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LENGTH     | 0                                   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| bitwise OR | 0                                   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| odd parity | 1 if number of ones is even, else 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | 0 |
| Header     | 0                                   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| hex        | 0                                   |    |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    | 1  |    |    |    |    |    |   |   | 0 |   |   |   |   |   |   |   |

图 4.

确定报头后, 在数组中填入零字节进行读取操作。

这将提供一个通过 MOSI 发送、通过 MISO 接收的字节数组, 其内容如下:

|      |           |      |      |      |      |      |      |      |           |      |      |      |
|------|-----------|------|------|------|------|------|------|------|-----------|------|------|------|
|      | Byte      | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Byte      | Byte | Byte | Byte |
|      | 0         | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8         | 9    | 10   | 11   |
|      | HEADER    |      |      |      |      |      | DATA |      |           |      |      |      |
| MOSI | 00        | 00   | 01   | 00   | 00   | 00   | 00   | 00   | 00        | 00   | 00   | 00   |
| MISO | DONT CARE |      |      |      | 00   | 00   | 01   | 00   | READ DATA |      |      |      |

ECHOED HEADER

图 5.

读取该寄存器的程序例程必须查看数据流的最后两个 32 位字。

倒数第二个 32 位字 (回显报头, 字节 4 至 7) 应等于通过 MOSI 线发送到 MACPHY 的命令报头, 而最后一个 32 位字则包含寄存器的实际数据。

请注意, 寄存器读取总是返回 32 位字的寄存器内容, 即使基础寄存器仅包含 16 位可用内容。

当由于使用无效的奇偶校验位计算方式导致报头格式错误, 或 SPI 传输错误造成报头出现奇偶校验错误时, 回显的控制报头及所有后续的 32 位字都将变为 0x40000000, 这表明存在“报头错误, HEADER BAD”状况。因此, MACPHY 将忽略该命令。连接的上位机可以利用该信息来检测控制命令 (Control Command) 错误, 并触发重新传输正确的控制命令。

## 寄存器写入示例:

假设您想要向 NCN26010 内部的 MAC Control 0 寄存器写入数据, 以便在集成的 MAC 与 PHY 之间启用 TX (发送) 和 RX (接收) 传输, 同时让 MACPHY 计算以太网帧的帧校验序列 (FCS), 以减轻 MCU 的计算负担。

请查阅 NCN26010 的数据手册以找到 MAC CONTROL0 寄存器。它位于 MMS 1 (内存映射选择组 1) 中, 地址为 0x0000。

要启用 TX 和 RX, 并且让 MAC 计算并自动附加 FCS (帧校验序列) 到它将发送的每个以太网帧上, 需要在寄存器中设置第 8 位 (FCSA)、第 1 位 (TXEN) 和第 0 位 (RXEN)。这将产生一个包含 0x00000103 值的待写入的数据字。

与上述寄存器读取示例类似, 我们首先需要确定单个寄存器写入的正确控制命令报头。按照上述方案, 我们可以得到:

|            |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |   |  |  |  |  |  |  |
|------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|---|--|--|--|--|--|--|
| Field      | 31                                  | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |   |  |  |  |  |  |  |
| WNR        | 0                                   | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |   |  |  |  |  |  |  |
| AID        | 0                                   | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |   |  |  |  |  |  |  |
| MMS        | 0                                   | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |   |  |  |  |  |  |  |
| ADDRESS    | 0                                   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |   |  |  |  |  |  |  |
| LENGTH     | 0                                   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |   |  |  |  |  |  |  |
| bitwise OR | 0                                   | 0  | 1  | 1  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |   |  |  |  |  |  |  |
| odd parity | 1 if number of ones is even, else 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | 0 |  |  |  |   |  |  |  |  |  |  |
| Header     | 0                                   | 0  | 1  | 1  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |   |  |  |  |  |  |  |
| hex        | 3                                   |    |    |    |    |    |    | 1  |    |    |    |    |    |    | 0  |    |    |    |    |    |    | 0  |   |   |   |   |   |   | 0 |   |   |   |  |  |  | 0 |  |  |  |  |  |  |

图 6.

确定报头后, 我们就可以在数组中填入要写入寄存器的数据。

这样我们就得到了一个字节数组, 通过 MOSI 发送, 通过 MISO 接收, 内容如下:

|      | Byte 0        | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8            | Byte 9 | Byte 10 | Byte 11 |
|------|---------------|--------|--------|--------|--------|--------|--------|--------|-------------------|--------|---------|---------|
|      | HEADER        |        |        |        | DATA   |        |        |        | Don't Care        |        |         |         |
| MOSI | 31            | 00     | 00     | 01     | 00     | 00     | 01     | 03     | 00                | 00     | 00      | 00      |
| MISO | DONT CARE     |        |        |        | 31     | 00     | 00     | 01     | 00                | 00     | 01      | 03      |
|      | ECHOED HEADER |        |        |        |        |        |        |        | ECHOED WRITE DATA |        |         |         |

图 7.

请注意, 奇偶校验的工作方式与寄存器读取示例相同。如果报头中的奇偶校验位出现错误, 则随后的 32 位字将被设置为 0x40000000。在这种情况下,

MACPHY 不会改变寄存器的内容。连接的 MCU 必须相应地处理这种情况, 使用完整的报头重新发送寄存器写入事务。此外, 重要的是要先发送 MSB 数据。

下面的代码片段提供了一个生成命令事务报头的函数实现示例：

```
struct TC6_command_header {
    unsigned int WNR :1;
    unsigned int AID :1;
    unsigned int MMS :4;
    unsigned int ADDR: 16;
    unsigned int LEN:7;
    unsigned int P:1;
};

unsigned int CalcCmdHeader(struct
TC6_command_header header) {
    unsigned int result = 0;
    result |= header.WNR<<29;
    result |= header.AID<<28;
    result |= header.MMS<<24;
    result |= header.ADDR<<8;
    result |= header.LEN <<1;
    result |= !_builtin_parity(result);
    return result;
}
```

图 8.

上述示例说明了如何生成有效的控制事务，以支持单个寄存器的读写操作。

有关如何在一次 SPI 传输中读取连续寄存器地址的详细信息，建议本应用笔记的读者查阅 OPEN 联盟 TC6 文档。

#### 发送和接收以太网帧

OA-TC6 协议的设计不仅允许通过 SPI 接口实现器件配置，而且还能利用同一接口以半双工或全双工方式双向传输以太网帧，具体取决于上位机系统上运行软件的功能而定。但请注意，混合数据段上的实际以太网通信始终是半双工的。

以太网帧以分块 (Chunk) 的形式进行双向传输。

#### 数据块 (Data Chunks) 的概念

NCN26010 执行 OA SPI 协议，通过“数据块” (data chunks) 与连接的上位机设备或系统传输以太网数据。

数据块是以太网帧的一部分，或者在较短的帧中，它可以是整个以太网帧。在发送数据时，数据块包含一个数据报头，其后是预定义数量的有效 (pay load) 数据。NCN26010 可以配置每个数据块的 pay load 数据为 8、16、32 或 64 字节，默认设置为 64 字节。当上位机从 NCN26010 MACPHY 接收数据时，首先发送 pay load 数据，之后跟随一个 32 位的数据报尾，报尾向上位机指示：

- 发送的数据中是否包含有效的以太网帧数据。
- 在上一次传输中是否发现了错误。
- NCN26010 接收缓冲区中准备读取的数据块的可用数量。
- 当前可用于 TX 数据传输的空数据块的数量。

如果整个以太网帧的长度超过了单个数据块所能传输的数据量，则需要将其切割成适合数据块的片段。只要 NCN26010 设备内部的 4 kbyte 发送和接收缓冲区有足够的空间，这一概念就支持在 NCN26010 和上位机设备之间中断以太网帧交换而不丢失数据。

以太网帧数据可能包含也可能不包含 4 字节 FCS (帧校验序列)，具体取决于如何配置。当配置为计算并自动附加 FCS 时，由上位机设备向 NCN26010 发送数据时预计不会附加 FCS。同样，当启用 NCN26010 内部的 FCS 检查时，输入帧无需提供 FCS，因为无效帧 (FCS 校验错误的帧) 将不会存储在 RX 缓冲区 (即被丢弃)。

NCN26010 可配置为存储转发 (Store and Forward) 模式或直通 (Cut Through) 工作模式。

在“存储转发”模式下，完整的数据帧会在被发送到单对绞线以太网数据段或从以太网媒介接收之前，全部存储在器件的缓冲区内。

直通模式对上位机上运行的软件的延迟和吞吐量性能有更严格的要求，因为上位机必须始终跟上以太网数据的输入和输出速度。在使用多任务操作系统时，长时间的中断服务例程或不适宜的任务切换会导致通信速度减慢，从而因 TX 缓冲区欠载 (underruns) 或 RX 缓冲区超限 (overruns) 而导致数据丢失。

本应用笔记将不涉及使用 NCN26010 的“直通”模式进行数据交换的内容，而是主要聚焦于“存储转发”模式的操作。

在“存储转发”模式下运行时，上位机需要时刻关注缓冲区的填充水平，尤其是 RX 端或下行 (downstream) 数据流。对于 TX 上行 (upstream) 数据流，上位机上运行的软件可以轻松地延缓数据发送，直到所需的足够数据块数量 (即，与当前计划传输的以太网帧大小相匹配的数量)。

由于上位机无法直接控制 RX (downstream) 数据流，若设备以半双工方式交换数据，应优先处理 RX 数据流。

除其他配置选项外，数据块大小 (chunk size) 的选择会对将 NCN26010 连接到上位机 MCU 或系统的 SPI 接口施加最低速度要求。

在使用简单配置时，只允许新的以太网帧从数据块的开头开始，所需的 SPI 速度取决于以太网帧的长度以及数据块的大小：以下图表说明了这种关系。

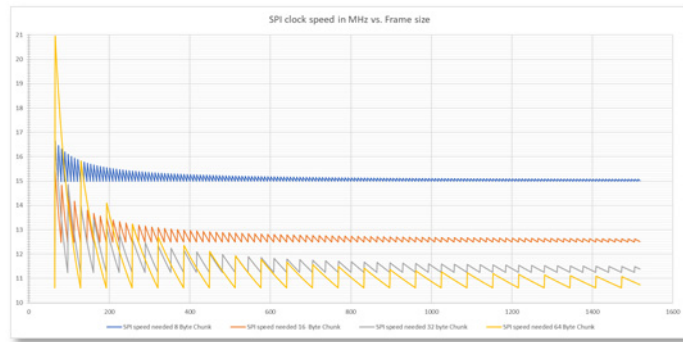


图 9.

从这张图表中，这种方法的局限性变得显而易见。当应用主要使用短帧时，根据 NCN26010 的配置方式，较小的数据块大小可能会带来更高的吞吐量。在这种简单的配置中，极端情况是使用长度为 65 字节的帧和 64 字节大小的数据块。这将需要 21 MHz 的 SPI 速度。

NCN26010 设计支持串行时钟 (SCLK) 上的 SPI 速度高达 25 MHz。因此即使在这种情况下，MACPHY 的速度也足以传输所需的数据；然而，应用程序或驱动程序必须同时处理 RX 和 TX。随着以太网帧大小的增大，默认的 64 字节数据块大小需要降低 SPI 接口的时钟频率，以保证数据传输。

在多点 (multi-drop) 传输环境中，我们可以假设大部分数据流都在 RX 方向。由于 10Base-T1S 的半双工特性，即使以半双工方式处理 SPITX 和 RX 数据流，25 MHz SCLK 也能在大多数情况下提供稳定的数据流而不会丢失数据。

当 NCN26010 需要在单对以太网数据段上传输数据时，上位机需要发送一系列包含完整以太网帧数据的有效数据块。

新的以太网帧将从数据报头的 SV 标志 (起始有效) 置 1 开始。在简单情况下，帧只允许在 CS (片选) 信号被置位后的数据块起始处开始，起始字偏移量 (Start Word Offset) 将始终为 0。如果以太网帧能够容纳在一个单独的数据块内，那么有效数据 (payload) 最后一个字节的 EV (End Valid, 结束有效) 标志也需要被设置。请注意，当要传输的以太网帧数据短于以太网帧长度 (包括源地址/目的地址、长度、字段/其他类型和

FCS) 的最小要求 64 字节时，则 NCN26010 内的 MAC 会自动添加填充字节，以将帧扩展到其最小所需长度。当 NCN26010 被配置为不自动添加 FCS 时，填充字节以及 FCS 才需要由上位机提供。

与控制事务报头的计算类似，数据事务报头也可按下面的示例代码生成：

```
struct TC6_data_header {
    unsigned int NORX :1;
    unsigned int DV :1;
    unsigned int SV: 1;
    unsigned int SWO:4;
    unsigned int EV: 1;
    unsigned int EBO: 6;
    unsigned int P:1;
};

unsigned int CalcDataHeader(struct TC6_data_header
header) {
    unsigned int result = 0x80000000; //DNC is always one
    for a data transaction
    result |= header.NORX<<29;
    result |= header.DV<<21;
    result |= header.SV<<20;
    result |= header.EV<<14;
    result |= header.EBO <<8;
    result |= header.SWO <<16;
    result |= !__builtin_parity(result);
    return result;
}
```

图 10.

### 发送以太网帧

在这里，我们仅描述了被称为“TX 帧在数据块边界结束”和“每个 TX 帧都匹配数据块”的 TX 帧发送情况。详见 [2]“图 8：发送数据块案例”。

我们需要区分以三种不同方式处理的帧数据情况：

1. 帧完全匹配一个数据块
2. 以太网帧匹配两个数据块
3. 以太网帧需要两个以上的数据块

在第一种情况下，用户需要发送一个数据报头，紧随其后的是实际的帧数据。该数据报头需要设置以下这些位：

- SV = 1
- DV = 1
- EV = 1

EBO (End Byte Offset, 结束字节偏移) 指向有效数据的最后一个字节。需要注意的是，即使以太网数据没有占满整个数据块 (例如，ARP 广播帧长度为 46 字节，所以默认的 64 字节数据块并不会被完全利用)，应用程序仍然需要始终发送全长数据块。

在使用 64 字节数据块且需要发送一个长度为 46 字节的 ARP 广播帧的情况下，EBO 将指向第 45 个字节，因为在数据块中的第一个字节索引为 0。

在第二种情况下，需要两个数据块来容纳 payload 数据，那么第一个数据报头需要设置：

- SV = 1
- DV = 1
- EV = 0
- EBO = 0
- SWO = 0

而帧中的第二个也是最后一个数据报头需要设置：

- SV = 0
- DV = 1
- EV = 1
- EBO = position of last valid pay load byte (最后一个有效 payload 字节的位置)
- SWO = 0

如果向 NCN26010 MACPHY 传输 payload 数据时需要两个以上的数据块，则第一个报头与前一种情况相似：

- SV = 1
- DV = 1
- EV = 0
- EBO = 0
- SWO = 0

在第一个数据块之后和最后一个数据块之前的所有报头都需要：

- SV = 0
  - DV = 1
  - EV = 0
  - EBO = 0
  - SWO = 0
- Or 0xA0200000.

最后一个报头需要设置：

- SV = 0
- DV = 1
- EV = 1
- EBO = position of last valid pay load byte (最后一个有效 payload 字节的位置)
- SWO = 0

由于在数据传输前要先传输报头，因此向 MACPHY 发送数据块的应用程序总是要比数据块的 payload 数据多发送 4 个字节。

由此，8 字节的数据块需要 12 字节，16 字节的数据块需要 20 字节，32 字节的数据块需要 36 字节，64 字节的数据块需要 68 字节。

请注意，即使数据块未全部使用，这些长度也是固定的。在 EV=1 数据块的 EBO 之后发送的数据为“不关心”数据，可以是任何随机数据。为方便使用，建议在数据块中填入全为 0 的字节。

与发送控制事务一样，数据事务也是双向同时进行的。

通过 SPI 接口向 NCN26010 发送数据时，上位机将通过 NCN26010 的 MISO (主入从出，Master In Slave Out) 引脚接收 MACPHY 返回的相同字节数。通常情况下，数据从一个足够大的缓冲区发送和接收，该缓冲区可容纳发送帧数据所需的所有数据块。输入的数据将进入第二个 (即接收) 缓冲区。

### 接收以太网数据

发送到 MACPHY 的数据在每个数据块之前带有 32 位报头信息，MACPHY 发送出的数据块在实际数据之后有一个 32 位报尾信息。

报尾包含了上位机用来引导进一步通信的信息，无论是发送还是接收更多数据。报尾的各字段在文献 [2] 的第 7.3.7 节中有详细解释。NCN26010 器件不提供 OPEN 联盟定义的可选时间戳功能，因此 RTSA (位 7) 和 RTSP (位 6) 字段将不使用，始终为 0。

NCN26010 器件的数据报尾结构如下：

|      |      |      |     |    |    |    |           |           |     |    |    |     |    |    |    |
|------|------|------|-----|----|----|----|-----------|-----------|-----|----|----|-----|----|----|----|
| 31   | 30   | 29   | 28  | 27 | 26 | 25 | 24        | 23        | 22  | 21 | 20 | 19  | 18 | 17 | 16 |
| EXST | HDRB | SYNC | RCA |    |    |    |           | DONT CARE |     | DV | SV | SWO |    |    |    |
| 15   | 14   | 13   | 12  | 11 | 10 | 9  | 8         | 7         | 6   | 5  | 4  | 3   | 2  | 1  | 0  |
| FD   | EV   | EBO  |     |    |    |    | DONT CARE |           | TXC |    |    |     | P  |    |    |

图 11.

这些字段可概括为以下功能：

|      |                                                                                                                       |
|------|-----------------------------------------------------------------------------------------------------------------------|
| EXST | 外部状态。当该标志为 1 时，上位机程序必须检查 SPI 协议 STA-TUS0 寄存器 (MMS: 0, 地址: 0x0008) 的内容，并根据错误情况进行相应处理。参见本应用笔记中的“错误处理”部分                 |
| HDRB | 报头错误。这表示 MACPHY 接收到奇偶校验位不正确的控制事务或数据块。因此，它表示数据丢失。上位机可以选择重新发送最后一个事务/数据块，或以其他方式处理这种情况。                                   |
| SYNC | 这是 SPICFG0 寄存器第 7 位的 Copy。它通常由上位机在完成 MACPHY 的配置后设置。如果 MACPHY 的 SYNCH 位被设置为 0，将不会发送或接收以太网数据。一旦 SYNCH 位被设置，只能通过复位器件来清除。 |
| DV   | 数据有效。其功能与发送数据相同。为 1 时，该数据块包含有效的以太网数据。                                                                                 |
| SV   | 起始有效。设置为 1 时，该数据块包含一个新以太网的起始部分。                                                                                       |
| SWO  | 起始字偏移。<br>指向数据块内新以太网帧起始的位置。在基本配置中，新帧数据只能在数据块的开始位置 (紧接着片选 Csn 有效信号之后) 起始，该字段应始终为 0                                     |

|     |                                                                                                                       |
|-----|-----------------------------------------------------------------------------------------------------------------------|
| FD  | 丢帧。只有在直通转发模式下才会出现。它指示上位机丢弃收到的整个以太网帧，主要原因是收到了错误的 FCS。                                                                  |
| EV  | 结束有效。在报尾设置了 EV 标志的数据块包含以太网帧的结尾部分。连接的上位机通常会读取数据块，直到看到报尾的 EV 标志。然后，它将完整的以太网帧传递给 TCP/IP 协议栈或任何其他处理 OSI 第 3 层及以上的应用软件或硬件。 |
| EBO | 结束字节偏移。与传输报头相同、当 EV 标志设置时，指向有效帧数据的最后一个字节。                                                                             |
| RCA | 可用接收数据块。告诉上位机，在 NCN26010 的接收缓冲区中还有多少有效数据块可供上位机读取。请注意，这些数据块可能属于不止一个完整的以太网帧。为避免接收缓冲区溢出造成数据丢失，上位机应尽量减少 RCA 数量。           |
| TXC | 可用传输数据块。告诉上位机有多少可用传输数据块可写入 NCN26010 的 TX 缓冲区。TXC 和 RCA 的值也可以通过轮询读取缓冲区状态寄存器 (MMS: 0, 地址: 0x000B) 来获取。                  |

以太网数据交换最简单但也最慢的方式是半双工方式。在发送帧时，如果在数据报头中设置了 NORX 标志，NCN26010 就会被指示在接收上位机发送的 TX 帧数据时，不向上位机发送以太网数据。

同样，当数据报头的 DV 标志设置为零时，上位机可在不提供 TX 数据的情况下接收帧数据。

在这种工作模式下，连接的上位机应该优先读取帧而非发送帧，或者以“公平共享”的方式交替进行发送和接收。建议是尽量快速地清空接收缓冲区。

### 基本配置

本节讨论了使用 NCN26010 所需的最低配置设置，以及一些可选特性。内容首先从无地址过滤的 CSMA/CD 简单情况开始，随后讨论如何启用其他附加功能：

- 地址过滤
- PLCA
  - ◆ 领导者 (Leader) 模式
  - ◆ 跟随者 (Follower) 模式
  - ◆ 突发 (Burst) 模式
- FCS 帧校验
- FCS 计算卸载
- 增强噪声抗扰度 (Enhanced Noise Immunity, ENI)

将 NCN26010 配置为最简单的操作形式，意味着将其设置为严格工作于 CSMA/CD (载波侦听多路访问/冲突检测) 模式，且不激活任何地址过滤选项。

首先，我们需要将 SPI 协议格式设置为所需的形式。在本应用笔记中，我们选择了以下 SPI 协议传输选项：

- 接收的帧将与接收数据块的 payload 起始位置对齐 (即字节 0)，紧跟上位机对片选 CSn 有效信号。
- 存储转发模式下的发送和接收
- 控制数据读/写保护禁用 (详见 [2] 第 7.4 节)
- 每个数据块的默认 payload 大小为 64 字节。

为了避免与无意中占用单对以太网 (Single Pair Ethernet) 介质的站点 (Station) 发生配置故障，建议按以下顺序配置设备：

#### 1. 器件重置

可通过三种不同方式强制器件复位

- a) 供电循环 (上电复位)；
- b) 使有效器件上的物理 RSTn 引脚；
- 或
- c) 写入配置寄存器。

由于有时错误处理需要器件复位，我们选择了后一种通过写入寄存器来复位器件的方法。NCN26010 有一个允许软复位的专用寄存器。在位于 MMS 0、地址 0x0003 的 RESET 寄存器中设置位 0 将会触发一次软复位。假设您有用来写入配置寄存器的例程，名为 T1SRegWrite (MMS、ADDR、DATA)。要发出器件复位指令，您需要调用 T1SRegWrite (0, 0x0003, 0x00000001)。

#### 2. PHY (物理层器件) 相关配置

- a.) PLCA
- b.) ENI

在这种简单的 CSMA/CD 配置中，除了通过将 PHYCTRL 寄存器 (MMS0，地址 0xFF00) 中的第 12 位设置为 1 来“激活链路”外，不需要任何操作：T1SRegWrite (0, 0xFF00, 0x00001000)

#### 3. MAC (媒体访问控制器) 相关配置：

- a.) FSC 过滤和计算
- b.) 地址过滤
- c.) 广播和组播过滤

对于基本操作，我们选择让设备自动添加 FCS (帧校验序列)，支持广播和组播消息，并且不过滤任何 MAC 地址。这实际上使器件运行在混杂模式下。为了实现这一点，我们在 MACCTRL0 (MMS 1，地址：0x0000) 中设置位 8 (FCS 添加)、位 1 (TX 使能) 和位 0 (RX 使能)：T1SRegWrite (1, 0x0000, 0x00000103)

#### 4. SPI (串行外设接口) 配置

- a.) 数据对齐
- b.) 直通转发或存储转发
- c.) 数据块 payload 大小
- d.) TXC 阈值
- e.) 最后设置 SYNC 位以启用通信

我们希望器件将所有新帧与片选有效信号 (CS) 对齐，从新的数据块的字节 0 开始。该器件应在 TX 和 RX 的“存储转发，S&F”模式下工作，并将 TXC 阈值设为 16。

所有设置均在 SPICFG0 寄存器中完成 (MMS 0，地址 0x0004)。

我们需要设置 SYNC (位 15)、CSARFE (位 13)、TXCTRSHESH = 0x3 (位 11 和 10)、CPS = 0x6 (位 2:0)。由此产生的寄存器内容为：0x0000AC06。

总之，按照下表所示的顺序写入寄存器 (我们称之为配置卡“card”)，即可实现基本操作。

最小 CSMA/CD 配置

| order | MMS | ADDR   | DATA       | comment                 |
|-------|-----|--------|------------|-------------------------|
| 1     | 0   | 0x0003 | 0x00000001 | Reset                   |
| 2     | 0   | 0xFF00 | 0x00001000 | Activate Link           |
| 3     | 1   | 0x0000 | 0x00000103 | Auto FCS + MAC enable   |
| 4     | 0   | 0x0004 | 0x0000AC06 | Config SPI and set SYNC |

请注意，上述列表显示的是复位后需要发送到器件的最基本信息，以便 MACPHY 能够通过单对以太网介质与上位计算机或 MCU 通信。

#### 增加 PLCA (物理层冲突避免)

当我们希望器件成为支持 PLCA 的网络的一部分时，我们需要配置并启用 NCN26010 MAC PHY 内部的 PLCA 调和子层 (RS) 的功能。有关物理层冲突避免 (PLCA) 的详细说明，请查阅 [1] 第 148 条。在启用 PLCA 的冲突域中，需要有一个所谓的“领导者”。这通常是被分配 PLCA ID=0 的器件。领导者是通过在线上发送 BEACON 信号来启动 PLCA 周期的站点。

领导者还需要配置一个最大节点数，该数必须等于或高于分配给本地冲突域中所有站点的最高 ID。

当我们希望器件成为跟随者节点时，必须为其分配一个本地 ID 并启用 PLCA。

具体方法是在 PLCACTRL1 寄存器中设置相应的节点 ID (位 7:0)，并在 PLCACTRL0 寄存器中启用 PLCA (位 15)。

例如，假设我们配置的节点选择了 ID=7。那么基于前一个示例的配置命令就需要扩展，现在将显示如下内容 (红色线条表示增加的配置项)：

最小 PLCA 跟随者模式节点配置

| order | MMS | ADDR   | DATA       | comment                 |
|-------|-----|--------|------------|-------------------------|
| 1     | 0   | 0x0003 | 0x00000001 | Reset                   |
| 2     | 0   | 0xFF00 | 0x00001000 | Activate Link           |
| 3     | 1   | 0x0000 | 0x00000103 | Auto FCS + MAC enable   |
| 4     | 4   | 0xCA02 | 0x00000007 | Set PLCA ID to 7        |
|       | 4   | 0xCA01 | 0x00008000 | Enable PLCA             |
| 5     | 0   | 0x0004 | 0x0000AC06 | Config SPI and set SYNC |

当我们要将节点配置为 PLCA 领导者模式时，本地 ID 需要设为 0，最大节点数需要设为一个合理的数字 (8 或更大)。

假设我们想设置一个领导者，以支持冲突域中的 8 个节点 (多点传送序列)。配置卡将更改为下图所示：

最小 PLCA 头节点 (节点数 8)

| order | MMS | ADDR   | DATA       | comment                         |
|-------|-----|--------|------------|---------------------------------|
| 1     | 0   | 0x0003 | 0x00000001 | Reset                           |
| 2     | 0   | 0xFF00 | 0x00001000 | Activate Link                   |
| 3     | 1   | 0x0000 | 0x00000103 | Auto FCS + MAC enable + Address |
| 4     | 4   | 0xCA02 | 0x00000800 | PLCA ID = 0 Node Count = 8      |
|       | 4   | 0xCA01 | 0x00008000 | Enable PLCA                     |
| 5     | 0   | 0x0004 | 0x0000AC06 | Config SPI and set SYNC         |

我们可以根据这一原则增加更多的配置选项。例如，PLCA 还提供了额外的功能，可用于处理应用中不同站点发送数据量不均的特定情况。例如，如果某个特定站点需要发送的数据量是其他所有站点的两倍，则可以允许它在每个 PLCA 传输机会中发送两帧或多于两帧，而不仅仅是一帧。这可以通过启用 PLCA 突发模式来实现 (见 [1] 第 148.4.4.2 条)。要启用突发模式，需将最大突发计数 (在单个传输机会期间可能额外传输的以太网帧数) 的 15:8 位设置为站点在每个传输机会中可发送的额外帧数。当希望站点在每个 TO (传输机会) 发送两帧时，我们需要将 PLCA BURST 寄存器中的 MAXBC 设置为 1。

配置 PLCA 头节点 (节点数 8) 并启用突发模式

| order | MMS | ADDR   | DATA       | comment                      |
|-------|-----|--------|------------|------------------------------|
| 1     | 0   | 0x0003 | 0x00000001 | Reset                        |
| 2     | 0   | 0xFF00 | 0x00001000 | Activate Link                |
| 3     | 1   | 0x0000 | 0x00000103 | Auto FCS + MAC enable        |
| 4     | 4   | 0xCA02 | 0x00000800 | PLCA ID = 0 Node Count = 8   |
|       | 4   | 0xCA05 | 0x00010080 | Allow one extra frame per TO |
|       | 4   | 0xCA01 | 0x00008000 | Enable PLCA                  |
| 5     | 0   | 0x0004 | 0x0000AC06 | Config SPI and set SYNC      |

#### 增强噪声抗扰度模式

在那些单对以太网 (Single Pair Ethernet, SPE) 线路受到较高噪声水平影响的应用场景中 (例如，靠近电机或交流线路，或是在单一多股双绞线内捆绑了多条 SPE 连接)，NCN26010 通过提供一个非标准特性——增强噪声抗扰度 (Enhanced Noise Immunity, ENI)——来展现出卓越的性能。ENI 在启用 PLCA 的数据段中工作时不会产生副作用。当在 CSMA/CD 工作模式中激活 ENI 时，站点冲突检测的能力会略微下降，但 ENI 并不会中断网络通信。

启用 ENI 可以通过简单地设置 PHYCFG1 寄存器 (MMS 4，地址 0x8001) 的第 7 位 (bit 7) 来完成。将此设置添加到之前的示例中，配置卡将如下所示：

配置 PLCA 头节点 (节点数 8)，启用突发模式，并开启增强噪声抗扰度 (ENI)

| order | MMS | ADDR   | DATA       | comment                      |
|-------|-----|--------|------------|------------------------------|
| 1     | 0   | 0x0003 | 0x00000001 | Reset                        |
| 2     | 0   | 0xFF00 | 0x00001000 | Activate Link                |
| 3     | 1   | 0x0000 | 0x00000103 | Auto FCS + MAC enable        |
| 4     | 4   | 0xCA02 | 0x00000800 | PLCA ID = 0 Node Count = 8   |
|       | 4   | CA05   | 0x00010080 | Allow one extra frame per TO |
|       | 4   | 0xCA01 | 0x00008000 | Enable PLCA                  |
|       | 4   | 0x8001 | 0x00000083 | Enable ENI                   |
| 6     | 0   | 0x0004 | 0x0000AC06 | Config SPI and set SYNC      |

### 地址过滤

接下来，我们可以探讨如何启用地址过滤功能。NCN26010 具备四个可配置的 MAC 地址过滤器，可以通过设置通配符，来匹配各种目标 MAC 地址或单个特定地址。

要启用地址过滤，首先要设置 MAC Control0 寄存器 (MMS 1, 地址 0x0000) 中的 ADRF 字段 (第 16 位)。然后在 ADDRFLT 寄存器中设置过滤规则，并在 ADDRMASK 寄存器中定义相应的过滤掩码。

如果器件要过滤特定地址，则相应 ADDRMASK 中的所有位都必须设置为 1。ADDRFLT 和 ADDRMASK 的长度均为 48 位，与标准以太网 MAC 地址的长度一致。只有当 MASK 位设置为 1 时，这些位才会被考虑用于过滤。

例如：地址过滤器只能存储与 MAC 地址 60:c0:bf:01:02:03 匹配的输入帧。

我们使用 ADDRFLT0H/L 将该地址输入过滤器：

ADDRFLT0H = 0x800060c0

ADDRFLT0L = 0xbf010203

请注意，ADDRFLT0H 中的第 31 位设置为 1 时将启用过滤规则。

由于我们希望过滤器查看整个 MAC 地址，因此需要相应地设置 ADDRMAK0H/L。

ADDRMAK0H = 0x0000FFFF

ADDRMASK0L = 0xFFFFFFFF

要正确设置过滤器和掩码，使用此功能的用户必须了解其工作原理：

过滤器会在以下情况下生效：如果输入以太网数据包的目标地址与过滤器寄存器的内容进行按位与 (bitwise AND) 运算后结果相等。

因此，

目标 MAC 地址 & ADDRMASK = 过滤器内容

这个等式必须成立，帧才能通过过滤器。

这样，如果我们希望过滤器只匹配 OUI (60:C0:BF 是 onsemi 的 OUI)

我们可以设置过滤器掩码为

ADDRMAK0H = 0x0000FFFF

ADDRMASK0L = 0xFF000000

并将过滤器设置为

ADDRFLT0H = 0x00060C0

ADDRFLT0L = 0xBF000000

如果我们对输入的目标地址 60:C0:BF:01:01:15 (此处使用二进制符号) 进行测试

0110 0000 1100 0000 1011 1111 0000 0001 0000 0001 0101  
&

1111 1111 1111 1111 1111 1111 0000 0000 0000 0000 0000  
=

0110 0000 1100 0000 1011 1111 0000 0000 0000 0000 0000  
6 0 C 0 B F 0 0 0 0 0 0

我们可以看到，MAC 地址过滤器接受所有目标地址以 60:C0:BF 开头的帧。

如果我们要在现有的配置基础上增加这个 MAC 地址过滤功能，配置如下：

配置 PLCA 头节点 (节点数 8)，启用突发模式，并开启增强噪声抗扰度 (ENI)、MAC 地址过滤功能  
60:C0:BF:01:02:03

| order | MMS | ADDR   | DATA       | comment                                    |
|-------|-----|--------|------------|--------------------------------------------|
| 1     | 0   | 0x0003 | 0x00000001 | Reset                                      |
| 2     | 0   | 0xFF00 | 0x00001000 | Activate Link                              |
| 3     | 1   | 0x0000 | 0x00010103 | Auto FCS +<br>MAC enable+<br>Filter Enable |
| 4     | 4   | 0xCA02 | 0x00000800 | PLCA ID = 0<br>Node Count = 8              |
|       | 4   | CA05   | 0x00010080 | Allow one extra<br>frame per TO            |
|       | 4   | 0xCA01 | 0x00008000 | Enable PLCA                                |
|       | 4   | 0x8001 | 0x00008003 | Enable ENI                                 |
|       | 1   | 0x0020 | 0xFF000000 | ADDRMASK0L                                 |
|       | 1   | 0x0021 | 0x0000FFFF | ADDRMAK0H                                  |
|       | 1   | 0x0010 | 0xbf000000 | ADDRFLT0L                                  |
|       | 1   | 0x0011 | 0x800060C0 | ADDRFLT0H +<br>enable bit set              |
| 6     | 0   | 0x0004 | 0x0000AC06 | Config SPI and<br>set SYNC                 |

所有这些例子都展示了如何使用配置寄存器，正如 NCN26010 数据手册中所述，来设置器件进入简单和高级工作模式。我们鼓励用户通过研究 NCN26010 数据手册来探索该器件更多的功能。

## 异常处理

以下是 NCN26010 向上位机发出错误或异常信号时可能采取的行动指南。本节内容应视为建议，因为根据应用需求，NCN26010 报告的错误或异常可能需要采取多种不同的操作。

每当 NCN26010 需要通知上位机发生的事件时，它将使 IRQn 引脚有效以在上位机上运行的软件中触发中断事件。

在没有错误或异常事件的正常情况下，触发中断是为了通知上位机已接收到新的以太网帧，并将其存储在 NCN26010 的 RX 缓冲器中，以备上位机接收。

每当触发中断时，OPEN 联盟在 [2] 中建议，通过由上位机获取有效的报尾信息，以确定是否需要采取除接收数据以外的其他操作。

如果上位机发现报尾中设置了 EXT 标志，则必须确定需要处理的错误或异常的类型。上位机可以通过读取 SPISTATUS0 寄存器的信息，来确定所报告的具体异常问题。

基于此，上位机可以从中提取 10 种不同的错误条件，我们将在下文中讨论。

状态寄存器可识别以下错误和异常：

| 符号     | 描述                                                                               |
|--------|----------------------------------------------------------------------------------|
| CDPE   | 控制数据保护错误。<br>仅当 SPI 协议配置为在控制命令保护模式下工作 (参见 [2]) 时，这将表明最后一个控制命令已被损坏。               |
| TXFCSE | 发送 FCS 错误。<br>仅当 MACPHY 被配置为期望从上位机接收 FCS 时，此错误表明通过 SPI 从上位机接收到的以太网帧已损坏。          |
| PHYINT | 由 NCN26010 内部集成的 PHY 检测到的异常。<br>上位机必须收集更多数据以确定根本原因。                              |
| HDRE   | 报头错误。<br>表示上位机发送的最后一个数据或控制报头出现奇偶校验错误。                                            |
| RESETC | 并非真正的错误。<br>每当 NCN26010 完成复位时，都会使 RESETC 有效。这是为上位机提供一个指示，检查并重新向 NCN26010 发送配置数据。 |
| LOFE   | 丢帧。<br>表示由于片选信号 (CS) 在事务 (无论是数据还是控制) 完全传输之前失效，导致事务未能完成。                          |
| RXBOE  | 接收缓冲区溢出错误。<br>NCN26010 内部的 4 kbyte 接收缓冲区已满。这意味着数据丢失。                             |
| TXBUE  | 发送缓冲区下溢错误。<br>仅发生在“直通转发”模式下，即上位机发送数据的速度慢于 MACPHY 在传输以太网帧时保持稳定数据流所需的速度。           |
| TXBOE  | 发送缓冲区溢出错误。<br>上位机试图向 NCN26010 MACPHY 发送数据，但其发送缓冲区已满。                             |

|      |                                                                                                                     |
|------|---------------------------------------------------------------------------------------------------------------------|
| TXPE | SPI 发送协议错误。<br>在报头中出现 TX 数据块协议错误，可能是以下情况之一：<br>在没有前一个 SV=1 的情况下，DV=1 重复出现 SV=1 而中间没有 EV。<br>SWO 或 EBO 指向超出数据块长度的位置。 |
|------|---------------------------------------------------------------------------------------------------------------------|

## 处理控制数据保护错误

当 SPI 状态 0 寄存器 (MMS 0，地址 0x0008) 的第 12 位被设置时，表示在控制命令事务的 SPI 传输过程中发生了数据保护错误。这仅在运行于控制命令保护模式时才会报告 (详情请参见 [2])，在该模式下，每个 32 位数据字后都有一个反码副本。

出现此错误时，NCN26010 会忽略控制命令。在这种情况下，上位机应重新发送控制命令。如果这些错误反复或频繁发生，则表明以下几种情况之一：

- SPI 速度太快
- SPI 引脚上的负载太高
- SPI 上位机 (host) 和下位机 (NCN26010) 之间的距离对于所选 SPI 速度来说太长
- PCB 上的噪声过大

在这种情况下，用户应验证其 PCB 设计并做出相应修改，或将 SPI 速度设为较低频率。

## 处理 TXFCS 错误

当 NCN26010 被配置为从上位机接收以太网帧数据的同时进行数据校验 (FCS，将 MACCONTROL0 寄存器的 FCSA 位设置为 0)，它会将输入的 FCS (帧校验序列) 与其内部计算的 FCS 进行比较。如果发现不匹配，它将发出 TXFCS 错误，通知上位机它从 SPI 接口接收到的最后一个以太网帧已损坏，并且不会通过单对线路发送给其他站点 (被废弃的帧)。

上位机应重新发送损坏的帧。

该错误如果频繁出现，可能表明：

- SPI 速度过高
- SPI 引脚负载过高
- SPI 引脚上的噪声太大

检查 SPI 接口上的信号完整性问题，并验证 SPI 接口上的时序。

## 处理报头错误

报头错误表示在通过 SPI 进行控制或数据传输时，报头中的奇偶校验位或其他任何位不慎被反转。

如果在控制数据传输中出现这种情况，上位机只需重新发送数据即可。在控制数据事务中，这种情况可在实际中断触发前被检测到，因为在这种情况下，回传的控制报头将接收 0x40000000。如果主机软件够响应回传的控制报头并检测到错误，就可以忽略中断。

当发生报头错误时，MACPHY 会忽略所有进一步的数据，直到 CSn 引脚失效。

上位机可以尝试重新发送帧，或依靠第 2 层以上的协议最终重新发送帧 (TCP 连接在未收到帧确认时会这样做)，或者完全忽略这种情况并接受数据丢失。

与之前的错误类型一样，我们建议用户检查其与 SPI 接口相关的软件和硬件实现。根本原因可能是接口时序太紧、接口速度太快、SPI 接口负载太高或其他各种问题。

#### 处理复位完成 (RESET Complete) 事件

当上位机从 STAUTS0 寄存器中提取到 RESETC 状态时，并不一定是通信出错了。

每次器件完成复位并准备就绪时，都会触发 RESET Complete (复位完成)。

例如，当设备上电或上位机使 RSTn 有效时，就会发生这种情况。

当 NCN26010 在不稳定的电源上运行时，工作电压降至欠压阈值以下，导致 MACPHY 复位时，也会发生这种情况。

在这种情况下，主机应重新配置 MACHPY。所有传输中的数据流以及存储在 RX 和 TX 缓冲器中的数据都将丢失。

#### 处理丢帧错误 (LOFE)

丢帧错误可能有两个原因：

1. 上位机和 NCN26010 之间 SPI 接口的信号完整性问题
2. 上位机上运行的软件未遵守 OA-TC6 SPI 协议。

在这两种情况下，器件都可能丢失数据。当 CSn (或 SS, slave select) 在 SPI 接口上的事务结束前被去电时，就会发生丢帧错误 LOFE。

要调试此问题，用户应监控 CS 线路，确保没有故障。

此外，还应检查软件例程是否符合 OA-SPI 规范 (参见 [2])。

当检测到 LOFE 时，上位机也可尝试重新发送数据。

#### 处理 RXBO 错误

顾名思义，当 NCN26010 的 RX 缓冲区溢出时，此标志会被置位。

在这种情况下，将丢失 RX 以太网帧数据。

为防止这种情况发生，上位机需要以足够快的速度运行 SPI，以应对 10 Mbit 的持续数据流量。请参阅“数据块概念”部分的图表以获得指导。

此外，上位机应优先处理 RX 流量。

请注意，NCN26010 没有在溢出情况下清空缓冲区的机制。

如果上位机发现溢出，最佳做法是停止 TX 数据流并尽快清空 RX 缓冲区。

#### 处理 TXBO 错误

当连接到 NCN26010 的上位机发送 TX 数据的速度超过 NCN26010 在线路上所能发送的速度时，就会发生发送缓冲区溢出错误。请记住，10Base-T1S 是一种多点数据段技术，在某些情况下，在单条总线上有多个 (有时多达 40 个) 潜在发送器。

避免发生这种错误的最简单方法是检查发送缓冲区中可用于写入的数据块数量。如果上位机中以太网帧比 TX 缓冲区中可用的数据块数量还要长，则应等待存储该帧所需的缓冲器数量可用后，再尝试将帧发送到 NCN26010 MACPHY。

一旦出现该错误，器件就已丢失数据。在大多数情况下，这将是发送的最后一个以太网帧，因此当这种情况发生时，上位机可以选择等待缓冲区有足够的可用空间，然后重新发送最后一个以太网帧，或者依靠上层机制来处理重新传输。

上位机可以通过查看 TXC 字段来轻松跟踪可用的缓冲区，该字段位于发送或读取缓冲区中的状态寄存器时，所接收的每个数据块报尾内。

在良好的情况下，TX 缓冲区溢出的情况永远不会发生。

#### 处理 TX 缓冲区下溢 (TXBU) 错误

当 NCN26010 设置为在直通转发模式下操作 TX 时，会标记 TXBU 错误。

尽管直通转发模式有助于缩短发送以太网帧的延迟时间，但上位机需要以 10 Mbit/sec 或更快的速度发送 TX 数据，以免在整帧发送到以太网介质之前 TX 缓冲区变空。

当发生 TXBU 错误时，至少会丢失最后一个以太网帧 (基本上是正在进行的传输)。

当 NCN26010 设置为存储转发模式时，可以完全避免 TXBU。

当应用依赖于直通转发操作时，上位机需要能够随时以足够快的速度发送以太网帧数据。所需的 SPI 频率在很大程度上取决于所发送的以太网帧的平均大小。

#### 处理 TXPE 错误

传输协议错误是 NCN26010 用户在软件开发期间可能会遇到的另一类错误。在准备部署的应用程序中，除了数据块被拒绝 (由于奇偶校验错误) 且未重新发送外，不应该出现 TXPE 错误。

TXPE 在违反基本 TC6 协议规范时触发。它们通常在以下情况下发生：

- NCN26010 在报头中看到一个起始有效，但没有之前的结束有效标志 (唯一的例外是设置 SYNC 位后的首次数据传输)
- DV = 1，但没有之前的 SV=1
- EV = 1，但没有之前的 SV=1
- SWO 或 EBO 指向超出数据块长度的位置

当发生这些错误时，客户应检查其软件实现，并修正数据流，使其符合 OA-SPI 协议 (见 [2])

#### 处理 PHYINT 错误

当 MACPHY 发出 PHYINT 事件信号时，需要采取额外的步骤。

为了确定事件的实际原因，上位机除了读取 SPISTATUS0 寄存器外，还必须读取 MIIM IRQ STATUS 寄存器 (MMS12，地址 0x0011)。

位 0 到 5 报告触发 PHYINT 事件的情况，这些事件包括：

- 物理冲突 (Physical Collision)
- PLCA 恢复 (PLCA Recovery)
- 远端 jabber (Remote Jabber)
- 本地 jabber (Local Jabber)
- PLCA 状态变化 (PLCA status change)
- 链路状态变化 (Link Status Change)。

关于如何处理这些事件没有明确的建议，因为大部分是信息性的，可能指向网络配置错误或噪声过大的问题。

有关 MIIM 中断条件的详细信息，请查阅 NCN26010 数据手册中的 MMS12 寄存器描述。

### 结语

在本应用笔记中，我们概述了 IEEE802.3cg 10Base-T1S 多点以太网 MACPHY NCN26010 的基本使用和配置。我们还就所需的 SPI 速度、如何设置可选和专有功能以及如何处理/避免错误和异常提供了指导。

onsemi, Onsemi, and other names, marks, and brands are registered and/or common law trademarks of Semiconductor Components Industries, LLC dba "onsemi" or its affiliates and/or subsidiaries in the United States and/or other countries. onsemi owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of onsemi's product/patent coverage may be accessed at [www.onsemi.com/site/pdf/Patent-Marketing.pdf](http://www.onsemi.com/site/pdf/Patent-Marketing.pdf). onsemi reserves the right to make changes at any time to any products or information herein, without notice. The information herein is provided "as-is" and onsemi makes no warranty, representation or guarantee regarding the accuracy of the information, product features, availability, functionality, or suitability of its products for any particular purpose, nor does onsemi assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. Buyer is responsible for its products and applications using onsemi products, including compliance with all laws, regulations and safety requirements or standards, regardless of any support or applications information provided by onsemi. "Typical" parameters which may be provided in onsemi data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. onsemi does not convey any license under any of its intellectual property rights nor the rights of others. onsemi products are not designed, intended, or authorized for use as a critical component in life support systems or any FDA Class 3 medical devices or medical devices with a same or similar classification in a foreign jurisdiction or any devices intended for implantation in the human body. Should Buyer purchase or use onsemi products for any such unintended or unauthorized application, Buyer shall indemnify and hold onsemi and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that onsemi was negligent regarding the design or manufacture of the part. onsemi is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

### PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:  
Email Requests to: [orderlit@onsemi.com](mailto:orderlit@onsemi.com)

onsemi Website: [www.onsemi.com](http://www.onsemi.com)

TECHNICAL SUPPORT  
North American Technical Support:  
Voice Mail: 1 800-282-9855 Toll Free USA/Canada  
Phone: 011 421 33 790 2910

Europe, Middle East and Africa Technical Support:  
Phone: 00421 33 790 2910  
For additional information, please contact your local Sales Representative