

Application Note

使用 **TPS65992S PD 控制器**控制 **TPS55288** 和 **TUSB1044**



摘要

本应用手册介绍了如何使用 TPS65992S USB Type-C®/PD 控制器 I2C 控制器功能来控制 USB-C 电力输送系统中的两个主要配套器件：TPS55288 降压/升压稳压器和 TUSB1044 线性转接驱动器（USB Type-C 交叉点开关）。本应用手册介绍了自动 I2C 控制所需的 TPS65992S 配置、TPS55288 用于管理 VBUS 电压的寄存器设置，以及 TUSB1044 用于处理 USB3/DisplayPort 通道开关的设置。本应用手册重点介绍了多字节 I2C 事务（例如，设置 TPS55288 输出电压基准）、对 TUSB1044 控制寄存器位（CTLSEL、FLIP_SEL、EQ_OVERRIDE 等）的解释，以及在连接、分离和 DisplayPort 交替模式事件期间保持正确的器件行为。文中以数据表为支撑对每个寄存器值进行了解释，以阐明配置背后的逻辑。用户将得到协调一致的设计，PD 控制器能够无缝协调电源和数据路径变化，而无需外部微控制器。

内容

1 简介.....	2
2 TPS65992S 应用工具中的 I2C 控制器配置.....	4
3 通过 I2C3 配置 TPS55288 降压/升压转换器.....	6
4 通过 I2C3 配置 TUSB1044 转接驱动器.....	10
5 I2C 事件表汇总.....	20
6 参考资料.....	22

商标

USB Type-C® is a registered trademark of USB Implementers Forum.
所有商标均为其各自所有者的财产。

1 简介

TPS65992S 是一款单端口 USB Type-C 和 USB 电力输送 (PD) 控制器，集成了各种数字接口，用于与其他器件进行通信。值得注意的是，TPS65992S 提供三个 I2C 端口：I2C1 和 I2C2 用作 I2C 外设接口（用于主机或 EC 通信），I2C3 用作 I2C 控制器。借助该 I2C 控制器端口，TPS65992S 可以直接控制板上的外部元件。在此处讨论的设计中，I2C3 用于管理两个关键器件：TPS55288 降压/升压转换器和 TUSB1044 线性转接驱动器。

TUSB1044 是一款 USB Type-C 交替模式转接驱动器开关（用于高速信号的线性中继器），支持高达 10Gbps 的 SuperSpeed 数据速率且不受协议影响。这意味着，在 DisplayPort 交替模式下，TUSB1044 可以传递 USB3.1 Gen2 信号，并通过 USB-C 连接器路由 DisplayPort 通道。可通过固定引脚设置或 I2C 接口来配置该器件以进行动态控制。内部寄存器可控制运行模式、高速通道映射翻转、均衡 (EQ) 设置和其他特性（有关详细信息，请参阅 TUSB1044 数据表寄存器映射）。使用 TPS65992S 作为 I2C 控制器，系统可以命令 TUSB1044 根据电缆方向和协商模式在仅 USB3 运行模式和各种 DisplayPort 交替模式配置之间切换。

表 1-1. TUSB1044 寄存器映射

偏移	首字母缩写词	寄存器名称
Ah	General_1	通用寄存器 1
Bh	General_2	通用寄存器 2
Ch	General_3	通用寄存器 3
10h	UFP2_EQ	UFP2 EQ 控制
11h	UFP1_EQ	UFP1 EQ 控制
12h	DisplayPort_1	AUX 监测状态
13h	DisplayPort_2	DP 通道启用/禁用控制
1Bh	SOFT_RESET	I2C 和 DPCD 软复位
20h	DFP2_EQ	DFP2 EQ 控制
21h	DFP1_EQ	DFP1 EQ 控制
22h	USB3_MISC	其他 USB3 控制
23h	USB3_LOS	USB3 LOS 阈值控制

TPS55288 是一款同步四开关降压/升压转换器，专为 USB 电力输送源应用而设计。TPS55288 可将输出电压调节至低于、等于或高于输入电压，从而支持宽输入电压范围（2.7V 至 36V）和 0.8V 至 22V 的输出电压范围。TPS55288 具有一个 10 位 DAC，用于设置决定输出电平的内部基准电压，其中 1LSB $\approx$ 1.129mV。通过 I2C 接口，可以对输出电压和电流限制进行编程，从而符合 USB PD 要求（包括可编程电源 (PPS)）。实际上，该转换器可以通过典型的 12V 电源提供高达 100W（例如 20V/5A）的功率。TPS55288 上的主要配置寄存器包括用于设置内部基准电压的 REF 寄存器（0x00 和 0x01）、用于电流限制的 IOUT_LIMIT 寄存器（0x02），以及用于压摆率、反馈选择等的寄存器。例如，REF 寄存器（0x00/0x01）形成一个 10 位值，用于对转换器基准电压进行编程；写入适当的值会根据内部反馈比调整输出电压。

表 1-2. TPS55288 寄存器映射

地址	首字母缩写词	寄存器名称
0h、1h	REF	基准电压
2h	IOUT_LIMIT	电流限值设置
3h	VOUT_SR	压摆率
4h	VOUT_FS	反馈选择
5h	CDC	电缆补偿
6h	模式	模式控制
7h	状态	运行状态

本应用手册介绍了如何配置 TPS65992S PD 控制器 I2C3 控制器接口，以协调方式控制 TPS55288 和 TUSB1044。我们将介绍 TPS65992S 应用程序自定义工具中的必要设置、I2C 命令序列的配置（使用正确的外设地址和寄存器地址）以及事件触发的 I2C 事务，使这些器件能够在 PD 协商和 USB-C 交替模式运行期间无缝运行。以下每个部分分别介绍了电源转换器和转接驱动器的设置，然后总结了此设计中使用的所有 I2C 事件。

2 TPS65992S 应用工具中的 I2C 控制器配置

为了使 TPS65992S 管理 TPS55288 和 TUSB1044，I2C3 控制器端口必须配置正确的外设地址以及命令索引到这些器件的映射。使用 TPS65992S 应用程序自定义工具（用于配置 PD 控制器固件的 GUI）、每个外部 I2C 外设都添加了 7 位地址，并分配了一个地址索引（PD 固件中使用的标识符）。该工具还允许定义与特定 PD 事件（如上电、连接、合约协商等）相关的 I2C 寄存器读取和写入操作序列。

在我们的设计中，I2C3 控制器端口设置如下：

- **TPS55288 (降压/升压转换器)** - 7 位 I2C 地址 **0x74**。该器件在 PD 控制器 I2C 控制器配置中添加为 **Address0** (地址索引 = 0)。我们在 PD 控制器 I2C 事件表中为针对 TPS55288 的命令保留寄存器索引 0 至 8。（注意：当器件 I2CADD 引脚/模式 位为 0 时，0x74 是 TPS55288 的默认 I2C 地址，这与我们的配置相匹配。）
- **TUSB1044 (USB-C 转接驱动器)** - 7 位 I2C 地址 **0x12**。此器件在 PD 控制器的配置中添加为 **Address1** (地址索引 = 1)。我们在事件表中为 TUSB1044 的命令分配寄存器索引 9 至 28。（I2C 模式下的 TUSB1044 地址由引脚搭接决定；此处的地址配置为 0x12。）

定义两个外设和地址后，TPS65992S 固件可以使用分配的地址索引将 I2C 事务引导至正确的器件。图 2-1 显示了 TPS65992S GUI 的示例：**Address0** 对应于 0x74 处的 TPS55288，**Address1** 对应于 0x12 处的 TUSB1044。通过地址映射，我们可以在 PD 控制器 I2C 控制器事件表中创建条目。每个条目（由上述寄存器索引编制索引）都指定一个事件触发、目标外设（通过地址索引）、该外设上的寄存器地址、要写入（或读取）的数据字节，以及命令是否与方向无关。在以下各节中，我们详细介绍了为每个器件配置的命令序列。

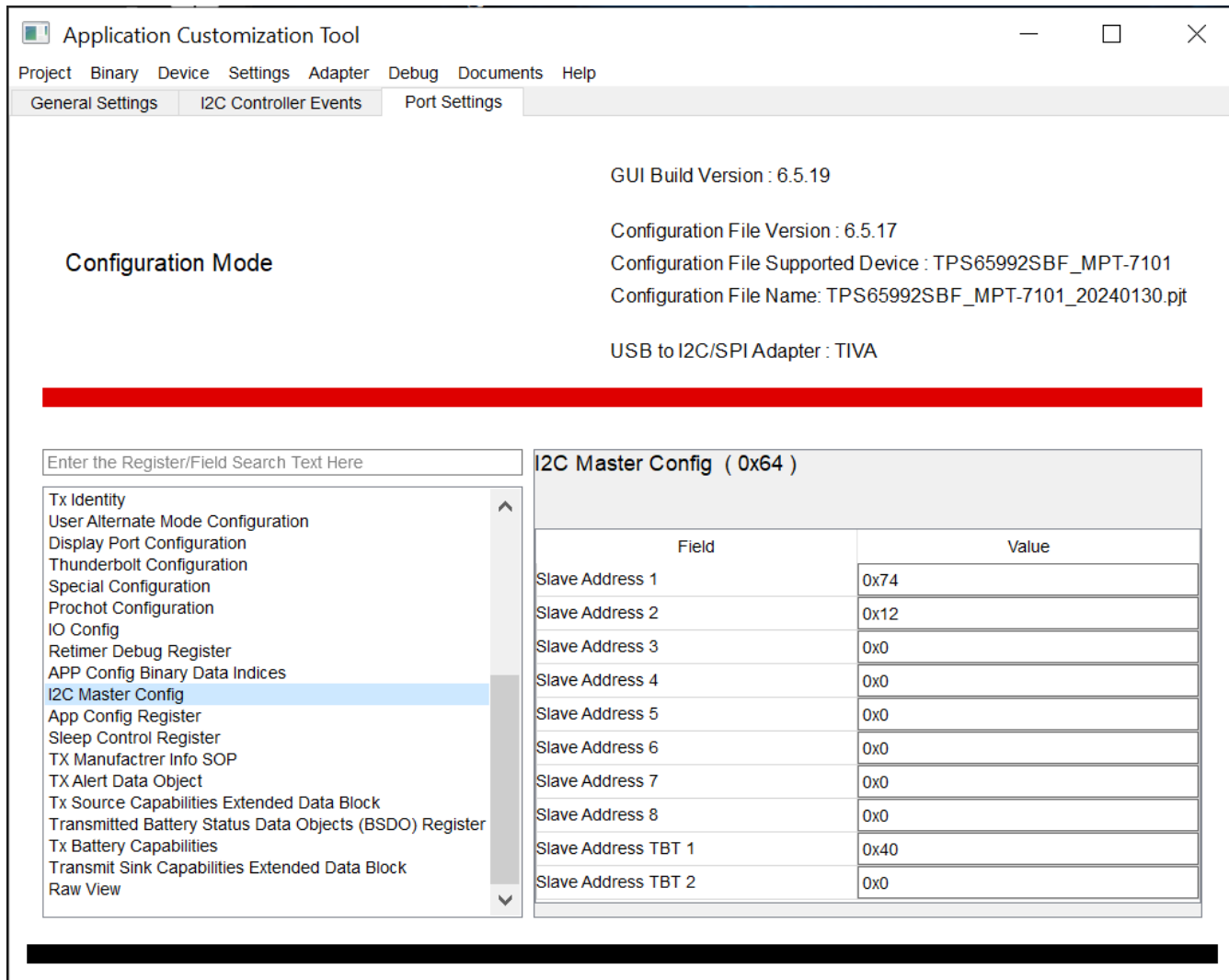


图 2-1. I2C 控制器配置上的 TPS65992S GUI 设置

3 通过 I2C3 配置 TPS55288 降压/升压转换器

对于 TPS55288，目标是根据 PD 协商对转换器输出行为进行编程，并在上电和断开时保持安全默认值。

TPS65992S 事件表 (索引 0 - 8 分配给 TPS55288) 会相应地进行设置。TPS55288 上的关键寄存器设置源自数据表：

- **REF (0x00/0x01)**：设置内部基准电压，该电压与反馈配置共同决定 V_{OUT} 。这是一个分为两个寄存器的 10 位值 (0x01 是 MSB, 0x00 是 LSB)。一个 LSB 对应于大约 1.129mV 的基准电压。通过调整 REF，可以将输出从大约 45mV 的最小基准电压 (0x0000，产生最低输出) 更改为 1.2V 基准电压 (0x03FF，产生最大输出)。复位后的默认 REF 值为 0x00D2 (LSB = 0xD2、MSB = 0x00)，对应于约 282mV。TPS55288 默认反馈比为内部检测设置 (请参阅下面的 V_{OUT_FS})，因此 282mV 基准在输出端产生约 5V 的电压 (详情见下文)。
- **V_{OUT_FS} (0x04)**：反馈选择寄存器控制转换器是使用内部电阻分压器进行 V_{OUT} 检测，还是使用外部反馈网络。反馈选择寄存器选择使用的内部反馈比。默认情况下， $V_{OUT_FS} = 0x03$ (二进制 0000_0011)，表示启用内部反馈 (位 7 = 0) 且内部反馈比设置为 0.0564 (位 1:0 = 11b)。0.0564 的比率对应于从基准到输出的 20 倍增益 (因为 $0.0564 \times 20 \approx 1$)，当基准最大值约为 1.129V 时，允许输出达到约 20V。更简单地说，使用默认比率时，282mV 基准产生大约 5V 的输出，1.129V 基准产生大约 20V 的输出，从而确定 PD 曲线所需的范围。
- **MODE (0x06)**：模式控制寄存器包含各种控制位；特别是位 7 是 OE (输出使能)，必须将其设置为 1 才能打开转换器输出。其他位配置特性，例如断续保护、关断时放电、I2C 地址选择和轻负载模式 (PFM/FPWM)。目标位 2 (I2CADD) 选择 I2C 地址 (0 表示 0x74，1 表示 0x75)，位 0 (MODE) 选择是从外部电阻器还是内部寄存器进行某些设置。在我们的设计中，我们保留 I2CADD = 0 (使用 0x74 地址) 并对 VCC 和 PFM 使用默认控制方法 (外部电阻器控制，位 0 = 0)，因为硬件设计提供适当的默认值。然而，我们会通过 I2C 启用输出 (OE)，并验证是否适当设置了其他位以实现运行。

使用这些寄存器时，TPS65992S I2C 事件序列配置如下：

上电初始化

在 PD 控制器的上电复位 (PoR) 上，我们将 TPS55288 初始化为已知的安全状态，以便 PD 控制器在端口激活后准备好默认提供 5V 电压 (标准 USB-C 默认电压)。在 PD 控制器复位后，TPS65992S 固件会立即触发对 TPS55288 的数次 I2C 写入：

- **设置基准电压**：将 0xD2 写入 TPS55288 寄存器 0x00 (REF 的 LSB)。这与默认 MSB (0x00) 结合使用，将 REF 值设置为 0x00D2。如上所述，0x00D2 对应于大约 282mV 基准，使用默认反馈比时，相当于大约 5V 输出。这样可以有效地对转换器进行 5V 输出编程，以此为起点。
- **设置反馈配置**：将 0x03 写入寄存器 0x04 (V_{OUT_FS})。这明确验证了转换器是否使用内部反馈网络和 0.0564 比率 (这是默认复位状态)。在此处写入 0x03 主要是为了完整性，这样可以在锁存任何非默认配置时确认反馈模式，或防止启动时出现任何不确定性。这会使内部反馈保持启用状态并选择比率，以通过内部调节允许完整的 PD 电压范围 (高达 20V)。
- **启用转换器输出**：将 0xA0 写入寄存器 0x06 (MODE)。这会设置位 7 (OE) = 1 以开启输出，并且位 5 = 1 (默认情况下保持断续模式处于启用状态)。值 0xA0 (二进制 1010_0000) 基本上会将 OE 翻转为 1，同时将其其他位保留为所需值 (例如，0x20 是 OE = 0、HICCUP = 1 等的复位值，0xA0 将 OE 更改为 1)。在进行此写入之后，TPS55288 将启用并调节至大约 5V 输出。

这三次写入 (写入 0x00、0x04、0x06) 在 PD 控制器初始化期间按顺序完成。通过在 PoR 时配置电源转换器，设计可以保持为即使在¹进行任何 USB-C 连接之前，TPS55288 也会处于默认的 5V 待机输出状态 (或在启用时准备提供 5V)。这对于符合 USB-C 标准非常重要，该标准要求 VBUS 上最初存在 5V 电压以进行连接。在我们的例子中，到连接器的 VBUS 实际启用由 PD 控制器电源路径开关控制，但 TPS55288 输出设置为正确的电平并在内部开启。

(选择的确切数值 0xD2、0x03、0xA0 对应于根据 TPS55288 数据表得出的所需启动配置。REF LSB 的 0xD2 (210 十进制) 是如文中所述的大约 5V 的代码， V_{OUT_FS} 的 0x03 选择具有 0.0564 比率的内部反馈，0xA0 启用输出。设计人员可以参考数据表中的 TPS55288 寄存器映射，以了解详细的位定义。)

动态 PDO 电压调整 (拉电流 PDO 协商)

转换器初始化后，PD 控制器接下来需要在每次协商新的 PD 协议时调整 TPS55288 输出。TPS65992S 配置为每当连接的器件选择拉电流 PDO 时（例如，只要 PD 源电压发生变化），就对 REF 寄存器发出 I2C 写入操作。在我们的示例中，假设系统提供四个分别为 5V、9V、15V 和 20V 的固定 PDO。我们在 TPS65992S 固件中定义了四个事件触发器。（例如，SRC_PDO1_NEGOTIATED、SRC_PDO2_NEGOTIATED 等等），每个都映射到向 TPS55288 REF 寄存器写入适当的值。

- **5V 合约 (PDO1)：**将 0xD2 0x00 写入 TPS55288 寄存器 0x00。这是两字节写入（0x00 是 REF LSB 的起始寄存器）。如前所述，数据字节 0xD2 0x00 设置 REF = 0x00D2（LSB = D2h、MSB = 00h），这对应于大约 5.0V 输出。（在 PD 工具中，此条目的数据长度 = 3：寄存器地址为 1 个字节 + 2 个数据字节。）
- **9V 合约 (PDO2)：**将 0x9A 0x01 写入寄存器 0x00（两字节数据，设置 REF = 0x019A）。在 TPS55288 内部 DAC 代码中，0x019A 对应于大约 9V。具体而言，0x019A = 410 十进制； $410 \times 1.129\text{mV} \approx 463\text{mV}$ 基准。使用 0.0564 反馈比时，可得出 $V_{\text{OUT}} \approx 463\text{mV} / 0.0564 \approx 8.21\text{V}$ 。但是，转换器输出会略微过冲以满足 9V 负载电压（确切值是根据校准和实验来选择）。根据典型设置，0x019A 是此设计中用于 9V 的代码。
- **15V 合约 (PDO3)：**将 0xC5 0x02 写入寄存器 0x00（REF = 0x02C5）。0x02C5 = 709 十进制； $709 \times 1.129\text{mV} \approx 800\text{mV}$ 基准。除以 0.0564，得到大约 14.2V。同样，考虑到负载和转换器容差，选择的代码在预期条件下产生大约 15V 的电压。
- **20V 合约 (PDO4)：**将 0xBF 0x03 写入寄存器 0x00（REF = 0x03BF）。0x03BF = 959 十进制； $959 \times 1.129\text{mV} \approx 1.083\text{V}$ 基准。使用 0.0564 比率时，可以得出大约 19.2V。目标是在空载或标称负载时达到约 20V（TPS55288 反馈和基准可以进行修整，使 0x03C0 可能过冲略高于 20V，因此使用 0x03BF 保持在典型值 20V 或略低于典型值 20V）。本质上，0x03BF 是我们设置中 20V 输入电压的校准代码。

其中每一个写入操作都针对 TPS55288 REF 寄存器，动态更改 V_{OUT} 。建立 PD 合约后，PD 控制器会立即发出相应的命令。这样一来，一旦接收端请求更高的电压（例如 15V），TPS65992S 就会写入新的 REF 值，而 TPS55288 会将输出转换为请求的电平。如果需要，输出的压摆率可由 TPS55288 VOUT_SR 寄存器 (0x03) 控制，但在我们的设计中，我们依赖转换器默认的压摆率，其通常配置为满足 PD 时序要求。

分离和复位行为

拔出 USB-C 电缆或分离接收端器件时，将电源恢复到安全状态。将 0xD2 0x00 写入寄存器 0x00 (LSB) 和 0x01 (MSB)。这种两字节写入操作会将基准电压完全复位为所需的 5V 默认设置，从而消除出现残余高压情况的可能性。无论如何，当断开时，TPS55288 输出会降至安全水平（约 5V 或更低）。这可以防止在拔下电缆后 VBUS 上出现任何残余的高电压。

通过为 TPS55288 设置这些 I2C 命令事件，所有电源调整均由 TPS65992S 固件自动执行，无需任何外部 MCU 干预。PD 控制器会监视 USB-C 状态和 PD 消息并触发适当的预编程 I2C 写入。这样可以保证，随着 PD 合约的变化，转换器无缝地提供请求的电压，当没有合约处于活动状态（分离或空闲）时，转换器处于已知的默认状态。

Record index 1 (0x1)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__POWER_ON_RESET
Data Length	3
Target Address Index	0
Orientation Independent	☐
Data	0xd200

图 3-1. 寄存器 0x00 上的上电复位事件

Record index 2 (0x2)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__POWER_ON_RESET
Data Length	2
Target Address Index	0
Orientation Independent	☐
Data	0x304

图 3-2. 寄存器 0x04 上的上电复位事件

Record index 3 (0x3)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__POWER_ON_RESET
Data Length	2
Target Address Index	0
Orientation Independent	☐
Data	0xa006

图 3-3. 寄存器 0x06 上的上电复位事件。

Record index 4 (0x4)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__SRC_PDO1_NEGOTIATED
Data Length	3
Target Address Index	0
Orientation Independent	☐
Data	0xd200

图 3-4. 寄存器 0x00 上的 SRC PDO1 协商事件

Record index 5 (0x5)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__SRC_PDO2_NEGOTIATED
Data Length	3
Target Address Index	0
Orientation Independent	☐
Data	0x19a00

图 3-5. 寄存器 0x00 上的 SRC PDO2 协商事件

Record index 6 (0x6)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__SRC_PDO3_NEGOTIATED
Data Length	3
Target Address Index	0
Orientation Independent	☐
Data	0x2c500

图 3-6. 寄存器 0x00 上的 SRC PDO3 协商事件

Record index 7 (0x7)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__SRC_PDO4_NEGOTIATED
Data Length	3
Target Address Index	0
Orientation Independent	☐
Data	0x3bf00

图 3-7. 寄存器 0x00 上的 SRC PDO4 协商事件

Record index 8 (0x8)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__DETACH
Data Length	3
Target Address Index	0
Orientation Independent	☐
Data	0xd200

图 3-8. 寄存器 0x00 上的分离事件

4 通过 I2C3 配置 TUSB1044 转接驱动器

TUSB1044 负责根据 Type-C 电缆方向和交替模式状态路由和调节高速信号 (USB3 SuperSpeed 通道和可选的 DisplayPort 通道)。通常, TUSB1044 可以在引脚控制 (GPIO) 模式下运行, 但在本设计中, 我们使用 *I2C 控制模式*, 通过 PD 控制器实现细粒度的动态控制。通过 TPS65992S I2C 控制器, TUSB1044 在发生各种事件时进行配置: 端口上电、电缆连接 (带方向) 和 DisplayPort 模式进入。所有与 TUSB1044 相关的 I2C 命令都分配给 PD 控制器表中的地址索引 1 (外设地址 0x12) 并占用事件索引 9 - 28。

我们配置中使用 TUSB1044 的主要寄存器包括:

- **General_1 (寄存器 0x0A)**: 该寄存器控制 TUSB1044 运行模式和特定覆盖操作。0x0A 中的重要字段:
 - 位 [1:0] **CTLSEL** - 选择数据模式: 0 = 禁用所有通道、1 = 仅启用 USB3、2 = 启用四个 DisplayPort 通道 (仅 DP)、3 = USB3 + 两个 DP 通道。
 - 位 [2] **FLIP_SEL** - 设置高速多路复用器的方向 (翻转): 0 = 正常方向、1 = 翻转方向。
 - 位 [4] **EQ_OVERRIDE** - 设置为 1 时, 允许使用 I2C 寄存器中的 EQ 设置 (覆盖引脚配置的 EQ 电平)。我们启用了此功能, 从而允许通过 I2C 对 EQ 进行编程。
 - 位 [5] **SWAP_SEL** - 如果设置, 则全局交换通道方向 (DFP 与 UFP)。在源端器件中, 我们通常将其保持为 0 (无交换), 除非设计需要反转方向 (例如, 用于接收端应用)。我们作为 DP 源端的配置保持 **SWAP_SEL** = 0 (默认值)。
 - 位 [3] **HPD_IN_OVERRIDE** - 未在我们的流程中使用 (控制 HPD 输入覆盖, 用于接收端应用)。
- **General_3 (寄存器 0x0C)**: 控制转接驱动器的 **VOD (电压输出差分)** 和 **直流增益** 设置, 以及用于在 I2C 模式下设置端口数据角色 (Dir) 的字段。重要字段:
 - 位 [6] **VOD_DCGAIN_OVERRIDE** - 1 以覆盖 VOD/直流增益引脚并使用寄存器设置。我们将其设置为 1, 以通过 I2C 对振幅进行编程。
 - 位 [5:2] **VOD_DCGAIN_SEL** - 4 位复合字段, 该复合字段会为所有通道选择 VOD 线性度和直流增益设置。这本质上对两个 2 位引脚设置 (CFG1 和 CFG0) 进行编码, 否则会设置 EQ 增益。在我们的设计中, 我们选择了可针对预期电缆长度提供适当信号振幅的设置。 (在提供的配置中, 写入的值对应于特定的引脚设置组合——请参阅下文)。
 - 位 [1:0] **DIR_SEL** - 设置器件数据角色: 0 = USB 和 DP 交替模式源端、1 = USB 和 DP 交替模式接收端、2 = USB + 自定义交替模式源端、3 = USB + 自定义交替模式接收端。我们的设计是源端 (例如, 笔记本电脑或输出 DisplayPort 的扩展坞), 因此我们使用 **DIR_SEL** = 0。
- **EQ 控制寄存器 (0x10、0x11、0x20、0x21)**: 这些寄存器可配置每个高速通道的均衡设置。TUSB1044 有四个高速差分对, 我们可以将其视为两个上行通道和两个下行通道, 每个通道都有 TX 和 RX 方向。在 I2C 模式下:
 - **0x10 : UFP1_EQ** - 位 7:4 设置上行端口通道 1 (UTx1) 的 TX EQ, 位 3:0 设置上行通道 1 (URx1) 的 RX EQ。
 - **0x11 : UFP2_EQ** - 同样, 用于上行通道 2 (UTx2 和 URx2)。
 - **0x20 : DFP1_EQ** - 位 7:4 用于下行端口通道 1 (DTx1, 转到连接器), 位 3:0 用于下行 RX1 (DRx1)。
 - **0x21 : DFP2_EQ** - 下行通道 2 (DTx2/DRx2) 的 EQ 设置。

通过写入这些寄存器, 我们可以微调每个通道的信号完整性。在我们的用例中, 我们对为 USB3 和 DP 信号推荐的某些 EQ 值进行编程 (我们配置中出现的值 0x66 和 0x33 对应于特定的 EQ 增益设置)。

下面介绍了如何进行 TPS65992S 配置, 以便在各种事件中控制 TUSB1044:

上电复位 (初始配置)

在 PD 控制器上电复位时, 我们希望 TUSB1044 在启用 I2C 控制的情况下以安全、禁用的状态启动。固件触发:

- 将 0x10 写入 TUSB1044 寄存器 0x0A。数据 0x10 (十六进制) 对应于二进制 0001_0000。查看位定义: 这将设置位 4 (EQ_OVERRIDE) = 1, 所有其他较低位 2:0 = 0 (CTLSEL = 000b, 这意味着禁用所有高速通道)。换句话说, 我们将启用 I2C EQ 覆盖, 但保持转接驱动器 TX/RX 路径关闭。这在复位时是一个很好的默认设置——转接驱动器在我们进一步配置转接驱动器之前不会转发任何高速信号, 并且转接驱动器依赖于寄存器来实现 EQ 设置, 而不是引脚。本质上, 0x10 会在软件控制下将 TUSB1044 置于空闲状态。 (复位时, TUSB1044 默认值也可以将通道禁用, 但写入 0x10 会明确验证模式和覆盖位是否正确。)

此命令会使 TUSB1044 保持未驱动任何信号，并准备好针对接下来的任何内容（USB 或交替模式）进行配置。这类似于在检测到电缆连接之前将转接驱动器保持在复位或待机状态。我们还根据需要对其他事件（如分离）使用相同的值以返回此基准。

Record index 9 (0x9)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__POWER_ON_RESET
Data Length	2
Target Address Index	1
Orientation Independent	☐
Data	0x100a

图 4-1. 寄存器 0x0A 上的上电复位事件

分离事件

在发生分离（电缆拔出）事件时，PD 控制器会再次将 0x10 发送至 TUSB1044 的寄存器 0x0A。这将写入与上电时相同的值：EQ_OVERRIDE = 1、CTLSEL = 0（通道关闭）。分离时执行此操作会有效地关闭转接驱动器通道切换并返回默认状态。任何有效的交替模式配置都将被清除，并且器件下次为新的连接序列做好准备。本质上，无论是在初始上电还是分离时，TUSB1044 都会被指示为禁用开关（无 USB3 或 DP 通道处于活动状态）并依靠 I2C 进行进一步配置。这可以防止在未连接任何器件时通过或锁存先前配置的不必要信号。

（注意：某些设计还可以在分离时切换 TUSB1044 硬件复位引脚。使用所示的 I2C 命令可以通过将控制寄存器写入已知的安全状态，无需额外的 GPIO 切换即可实现类似的结果。）

Record index 10 (0xa)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__DETACH
Data Length	2
Target Address Index	1
Orientation Independent	☐
Data	0x100a

图 4-2. 寄存器 0x0A 上的分离事件

连接事件 - 电缆方向处理

连接 USB-C 电缆后，TPS65992S 会检测方向（通过 CC 引脚），并启动 USB-USB PD 协商。在进入交替模式之前，即时的任务是为 TUSB1044 内的 SuperSpeed 多路复用器配置正确的方向，以便 USB3 信号（和未来的 DP 信号）能够正确路由。TPS65992S 固件区分了两个方向情况，通常标记为 ATTACH_UU 和 ATTACH_UD（这些标签来自 PD 控制器事件定义 - 本质上是指上行端口上行和下行端口下行方向，或者等效电缆未翻转和电缆翻转）。更简单地说：

- ATTACH_UU 对应于连接器 A 侧朝上的方向（通道的一个特定映射），
- ATTACH_UD 对应于旋转 180° 的连接器（反向通道映射）。

对于每个方向，我们都定义了对 TUSB1044 的 I2C 写入序列，以设置正常 USB3 运行（并准备潜在的 DP 模式）：

Attach_UU 序列：（默认方向的电缆）PD 控制器快速连续发出以下写入：

1. 将 0x11 写入寄存器 0x0A : 这会将 General_1 寄存器设置为 0x11 十六进制 (0001_0001 二进制)。与 0x10 相比, 现在有位 0 = 1 (CTLSEL = 001b), 同时保持位 4 = 1 和位 2 = 0。CTLSEL = 1 表示启用了仅 USB3.1 模式。因此, 0x11 会在给定方向下通过转接驱动器打开 USB3 路径。此处的位 2 (FLIP_SEL) 为 0, 表示采用正常方向 (因为这是 UU 情况)。因此, 0x11 将 TUSB1044 配置为 USB3 模式, 而不是翻转。

解释说明 : TUSB1044 将 SuperSpeed 发送和接收通道从主机侧连接到连接器上与非翻转插头插入相对应的 TX/RX 引脚。此时, DP 通道 (如有) 保持禁用状态 (因为 CTLSEL = 1 会选择仅 USB3)。EQ_OVERRIDE 保持为 1, 因此我们使用编程的 EQ 设置 (在后续步骤中设置)。

1. 将 0x58 写入寄存器 0x0C : 我们将 0x58 写入 General_3。二进制格式的 0x58 = 0101_1000。分解 :
 - 设置位 6 = 0x40 (0x58 包含 0x40), 因此 VOD_DCGAIN_OVERRIDE = 1 (使用寄存器进行 VOD/DC VOD 设置)。
 - 位 5:2 = 0b0110。根据数据表, 该字段 [5:2] 编码 CFG1 和 CFG0 引脚设置等效的值。0b0110 对应于 CFG1 = 01 (R) 和 CFG0 = 10 (F)。换句话说, 我们选择了一个特定的 VOD/直流增益电平 (一个可以对应于中等 EQ 设置—R 和 F 可能表示电阻器和悬空组合)。这可能是根据 TUSB1044 数据表建议或实验室调优而确定的。
 - 位 1:0 = 0b00, 设置 DIR_SEL = 0, 表示源模式 (USB + DP 源)。这与我们的系统角色相匹配 (TUSB1044 将上行侧视为主机侧, 将下行侧视为连接器侧)。

因此, 0x58 本质上表示: 使用 I2C 提供的 VOD/直流增益, 应用特定的增益设置 (R-F 配置), 并确认器件是源端。我们在连接时编写此文件, 以保证针对即将发生的信号正确配置转接驱动器的输出电平。

1. 将 0x66 0x66 写入寄存器 0x10 : 这是对 UFP1_EQ 寄存器 (0x10) 的两字节写入。数据 0x66 0x66 会将 UFP1_EQ 的高四位和低四位配置为 0x6。具体来说, 0x10 控制了一对 UFP TX/RX EQ :
 - 对于 UFP 通道 1, 0x66 作为单字节表示 TX EQ = 0x6 且 RX EQ = 0x6。我们发送两个字节 0x66, 这可能意味着我们也打算以单个多字节序列 (0x10, 后跟 0x10 和 0x11 的数据) 中的下一个寄存器 (0x11) 为目标。不过, 配置列表显示 0x66 0x66 到 0x10, 这实际上可以一次性加载 0x10 = 0x66 和 0x11 = 0x66 (具体取决于 PD 工具如何构建多字节写入帧)。

本质上, 我们会将 TX 和 RX 的所有上行 EQ (两个通道) 设置为中值 (0 × 6 是中程设置)。这可以是运行 USB3 或 DP 时转接驱动器主机控制器侧的建议 EQ 设置。这样会覆盖基于引脚的默认 EQ 并验证信号质量 (补偿电路板迹线损耗等)。

1. 将 0x33 0x33 写入寄存器 0x20 : 另一个两字节写入, 这次写入到 DFP1_EQ (0x20)。0x33 0x33 同样将 TX 和 RX 的下行 (连接器侧) 通道 1 (如果是两个字节, 则可能为通道 2) EQ 设置设为 0x3。值 0x3 的增益略低于 0x6——这可能表明转接驱动器与连接器的连接较短或表示电缆的默认要求。本质上, 我们也对两个通道的下行 (USB-C 连接器) EQ 进行编程。

上述 EQ 设置的组合 (UFP 侧 0x6、DFP 侧 0x3) 可能由 TI 的指南或实验室调优决定, 以便通过给定 PCB 和连接器的 USB3 信号完整性和合规性测试。

这四次写入 (0x0A、0x0C、0x10、0x20) 完成了 Attach_UU 配置。此时, 对于非翻转电缆插入, TUSB1044 设置为活动 USB3 模式: 连接正确的高速通道并调整 EQ/增益。

Attach_UD 序列 : (电缆翻转方向) 当电缆反转时, PD 控制器会触发 ATTACH_UD 事件。序列非常相似, 但方向不同时会有所不同 :

1. 将 0x15 写入寄存器 0x0A : 0x15 十六进制 = 0001_0101 二进制。与 0x11 (0001_0001) 进行比较 :
 - 位 2 (FLIP_SEL) 现在为 1 (因为 0x15 设置了位 2, 即 0x15 = 21 (十进制), 包括 0x04)。
 - 位 1:0 仍为 01 (CTLSEL = 1, 仅 USB3 模式)。
 - 位 4 保持为 1 (EQ_OVERRIDE 开启)。

因此, 0x15 配置启用仅 USB3, 但方向翻转。这会通知 TUSB1044 相应地路由通道 (相对于器件内部多路复用器交换 A 端口通道和 B 端口通道)。本质上, 该器件将 USB3 信号连接到与 UU 情况相比的一组高速引脚, 以解决翻转的电缆问题。

2. **0x58 至 0x0C** : **General_3** 寄存器获取与之前相同的 0x58 值。这实际上与方向无关——VOD/直流增益设置和 DIR (源端) 无论翻转如何, 都保持不变。因此, 我们对 UU 和 UD 使用相同的 0x58 配置。(这很有道理: 翻转方向不会改变我们是具有特定 EQ 偏好的源端这一事实; 这只会改变哪些物理通道承载信号。)
3. 将 **0x66 0x66** 写入寄存器 **0x10** : 我们再次将相同的 EQ 设置写入 UFP EQ 寄存器。在这里, 我们必须考虑: 在翻转方向上, 哪些 TUSB1044 通道对应于哪个物理连接? 该器件会交换通道 1 与通道 2 的通道使用情况, 如 FLIP_SEL 所示。但是, 通过将相同的值写入两个 UFP 通道 (0x10 和 0x11 = 0x66, 通过采用两字节序列), 我们可以验证无论哪个通道成为 USB 的实际 TX/RX, EQ 都设置为 0x6。换句话说, 在我们的配置中, 两个上行通道都具有相同的 EQ, 因此翻转不需要因对称性而需要不同的值。
4. 将 **0x33 0x33** 写入寄存器 **0x20** : 同样, 我们将两个下行通道的 EQ 设置为 0x3 (0x20 和 0x21 均通过多字节写入获得 0x33)。这种对称性意味着 UD 情况在两个可能的连接器方向上使用相同的 EQ 强度。

结果是 UD 序列仅在 0x0A 寄存器值 (0x15 与 0x11) 中不同。所有其他寄存器写入 (0x0C、0x10、0x20 和数据) 在 UU 和 UD 连接之间是相同的。这很有道理: 唯一改变的是翻转位, 它告诉转接驱动器以哪种方式路由通道。通过比较这两者, 我们可以看到 0x0A = 0x11 与 0x15 之间存在 0x04 的差异, 两者确实对应于 FLIP_SEL 位。其他所有内容 (EQ 增益等) 都保持不变, 表明我们的设计不需要针对翻转和未翻转设置不同的增益设置, 而是以相同的电气方式进行处理。

连接事件序列 (UU 或 UD) 后, TUSB1044 将配置为正常 USB3 操作。如果未启动交替模式, 系统将继续在 USB3 流量流经转接驱动器的情况下运行。连接时的 PD 控制器工作基本上已完成: 这可以保持高速路径方向正确并得到优化。

Record index 11 (0xb)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__ATTACH_UU
Data Length	2
Target Address Index	1
Orientation Independent	☐
Data	0x110a

图 4-3. 寄存器 0x0A 上的连接事件

Record index 12 (0xc)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__ATTACH_UU
Data Length	2
Target Address Index	1
Orientation Independent	☐
Data	0x580c

图 4-4. 寄存器 0x0C 上的连接事件

Record index 13 (0xd)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__ATTACH_UU
Data Length	3
Target Address Index	1
Orientation Independent	☐
Data	0x666610

图 4-5. 寄存器 0x10 上的连接事件

Record index 14 (0xe)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__ATTACH_UU
Data Length	3
Target Address Index	1
Orientation Independent	☐
Data	0x333320

图 4-6. 寄存器 0x20 上的连接事件

Record index 15 (0xf)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__ATTACH_UD
Data Length	2
Target Address Index	1
Orientation Independent	☐
Data	0x150a

图 4-7. 寄存器 0x0A 上的连接事件

Record index 16 (0x10)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__ATTACH_UD
Data Length	2
Target Address Index	1
Orientation Independent	☐
Data	0x580c

图 4-8. 寄存器 0x0C 上的连接事件

Record index 17 (0x11)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__ATTACH_UD
Data Length	3
Target Address Index	1
Orientation Independent	☐
Data	0x666610

图 4-9. 寄存器 0x10 上的连接事件

Record index 18 (0x12)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__ATTACH_UD
Data Length	3
Target Address Index	1
Orientation Independent	☐
Data	0x333320

图 4-10. 寄存器 0x20 上的连接事件

DisplayPort 交替模式配置

如果连接的伙伴器件（接收端器件，例如监视器或扩展坞）支持 DisplayPort 交替模式，并且 PD 协商进入该模式，则会触发额外的 I2C 事件以重新配置 TUSB1044，从而实现 DisplayPort 运行。在我们的 PD 控制器配置中，我们定义了与特定 *DisplayPort* 配置步骤相对应的事件，为方便起见标记为 *DP_CONFIG_ACE* 和 *DP_CONFIG_BDF*，每个都有方向的变体（UU 或 UD）。这些标签指的是标准 USB-C DisplayPort 引脚分配——通常是引脚分配 C 和 E 与 B、D 和 F。本质上，PD 控制器会根据四个高速通道用于 DisplayPort 的方式使用不同的配置序列。一种配置可以适用于 4 通道 DisplayPort（无 USB3 数据），另一种配置可以适用于 2 通道 DisplayPort + USB3 运行，这是 DP 交替模式的共模。确切的命名（ACE、BDF）源于这样一个事实：源端可以广播某些引脚映射的支持（例如，4 通道的引脚分配 C 或 E、2 通道 + USB 的引脚分配 D 或 F）。我们的配置可确保在每种情况下正确设置 TUSB1044 以支持通道布线和 EQ。

我们总结了这些序列：

- *DP_CONFIG_ACE_UU*：该事件对应于进入 DisplayPort 交替模式配置，该配置使用非翻转方向的（A、C、E）映射。在此模式下，通常所有四个通道都专用于 DP（USB3 被丢弃）。PD 控制器写入：
 - *0x1A 至 0x0A*：0x1A 十六进制 = 0001_1010 二进制。位 [1:0] = 10 (CTLSEL = 2)，根据 TUSB1044，这表示已启用 4 个 DisplayPort 通道。位 2 = 0（UU 方向），位 4 = 1（EQ 覆盖仍开启）。因此，0x1A 会将转接驱动器以正常方向切换为 DP 模式（所有通道均携带 DP）。
 - 将 0x55 0x55 写入寄存器 0x10：对 UFP1_EQ（和 UFP2_EQ）的两字节写入。数据 55 55 将每个上行 EQ 寄存器设置为 0x55。中断 0x55：每个通道上的 TX 为 0x5，RX 为 0x5。因此，在 DP 模式下，上行侧的 EQ 设置略有不同：0x5（比我们用于 USB 的 0x6 低一位）。这可以优化 DP 信号质量（因为 DP 信号可能具有不同的均衡需求，或者 DP 源端可以执行一些 EQ）。

在此事件中，我们没有明确看到向 0x20 写入值（这可能表明下行 EQ 保持为设置状态，或者 4 通道 DP UU 不需要更改）。假设下行（面向连接器）EQ 可以保持在最后设置的任何值（连接时为 0x33）。这是合理的，因为无论是承载 USB 还是 DP，连接器的物理通道特性都不会改变；但是，也可以针对 DP 频率进行调整。在我们的配置表中，ACE_UU 未列出新的 0x20 写入，因此我们在 DFP EQ 上保持与之前相同的 0x33 0x33。

- *DP_CONFIG_ACE_UD*：这是与上述方向相对应的翻转方向：

- **0x1E 至 0x0A** : 0x1E 十六进制 = 0001_1110。位 [1:0] = 10 (CTLSEL = 2, 仍为 4 通道 DP), 位 2 = 1 (翻转方向), 位 4 = 1 (EQ 覆盖开启)。因此 0x1E 表示 **DP 4 通道模式, 翻转方向**。
- **将 0x55 0x55 写入寄存器 0x10** : 将相同的 55 55 数据写入 UFP EQ 寄存器。我们再次看到, ACE_UU 和 ACE_UD 之间的唯一区别是 0x0A 值 (0x1A 与 0x1E)。0x04 差异表示翻转位, 与连接事件完全相同。寄存器 0x10 的数据保持为 0x55 0x55 (由于翻转, 因此未发生变化, 因为我们再次将两个通道设置为相同的值)。

和以前一样, ACE_UD 也未显示新的 DFP EQ 写入, 这意味着我们序列中的这个 DP 模式条目不会改变 DFP 侧 EQ。

- **DP_CONFIG_BDF_UU** : 这对应于不同的引脚分配场景——可能是 2 个 DP 通道 (B 和 D) + USB3 (F)。在这种情况下, PD 控制器写入 :

- **0x1B 至 0x0A** : 0x1B 十六进制 = 0001_1011。位 [1:0] = 11 (CTLSEL = 3), 这意味着 **USB3 + 2 通道 DP 模式**。位 2 = 0 (无翻转, UU), 位 4 = 1 (EQ 覆盖开启)。因此, 0x1B 将 TUSB1044 配置为路由 DP 的四个通道中的两个通道, 并在正常方向上为 USB3 保持一对通道。例如, 这对应于对 DP 使用两个通道 (HBR 通道), 并将其他两个通道用于 USB3 TX/RX。
- **将 0x66 0x55 写入寄存器 0x10** : 我们有一个连接 UFP EQ 寄存器的两字节序列, 其中包含数据 66 55。这与以前的情况不同。可能的含义 : UFP1_EQ = 0x66、UFP2_EQ = 0x55 (由于第一个字节 0x66 进入寄存器 0x10, 第二个字节 0x55 进入寄存器 0x11)。实际上, 我们将一个上行通道 EQ 设置为 0x6, 而另一个设置为 0x5。为什么 ? 因为在 BDF 场景中, 一个上行通道传输 USB3 (我们之前调优为 0x6), 而另一个通道传输 DP (我们确定 0x5 已足够)。因此, 通道 1 (例如仍用于 USB3 的通道 1) 可以使用 EQ = 6, 而通道 2 (承载 DP 通道) 使用 EQ = 5。0x66 0x55 的顺序表明 : UFP1_EQ = 0x66 (对于 USB3 通道), UFP2_EQ = 0x55 (对于 DP 通道)。
- **将 0x33 0x33 写入寄存器 0x20** : 以 33 33 对 DFP1_EQ (0x20) 和 DFP2_EQ (0x21) 进行两字节写入。无论携带的信号如何, 这都会将下行通道 EQ 设置为 0x3 (与之前相同)。有趣的是, 该表意味着在 BDF_UU 中, 0x20 没有更新 (即使这是与连接事件相同的值)。或许这样做只是为了保证 DP 通道输出 (现在可能占用不同的物理线路) 设置为已知 EQ。在任何情况下, DFP1_EQ 和 DFP2_EQ 都为 0x33, 两者对称, 相对于先前的值保持不变。

- **DP_CONFIG_BDF_UD** : 双通道 DP + USB 场景的翻转情况 :

- **0x1F 至 0x0A** : 0x1F 十六进制 = 0001_1111。位 [1:0] = 11 (USB3 + 2DP)、位 2 = 1 (翻转)、位 4 = 1 (覆盖开启)。因此, 0x1F 本质上为 0x1B 加上翻转位 (0x04), 即 “USB3 + 2DP, 呈翻转方向”。
- **将 0x55 0x66 写入寄存器 0x10** : 数据 55 66 是从 0x10 开始写入。这可能会设置 UFP1_EQ = 0x55 和 UFP2_EQ = 0x66。为什么与 UU 情况相比交换字节 ? 因为在翻转时, 承载 USB 与 DP 的两个上行通道的作用可以交换。在 UU 中, 我们假设通道 1 是 USB, 通道 2 是 DP (导致通道 1 为 66, 通道 2 为 55)。在 UD 方向上, 物理布线会导致相反的分配——通道 1 最终会承载 DP, 而通道 2 承载 USB3 (取决于通道的翻转方式)。PD 配置通过交换 EQ 设置来预测这一点 : 现在通道 1 获得 0x5, 通道 2 获得 0x6。这可保证任何处理 USB3 的通道都具有较高的 EQ (0x6), 处理 DP 的通道具有 0x5, 即使翻转时也是如此。BDF_UU 和 BDF_UD 配置之间的显式差异 (66 55 与 55 66) 突出了方向翻转如何需要交换特定于通道的设置。
- **将 0x33 0x33 写入寄存器 0x20** : 与 UU 一样, 我们将 0x33 0x33 写入 DFP EQ 寄存器。在 BDF_UD 中, 这可能是相同的操作 : 将两个下行通道保持在 EQ = 3。由于翻转并不能真正改变两个连接器通道 (其中两个承载 DP, 另一个承载 USB, 可能还有一个未使用) 具有相同的 EQ 这一事实, 因此我们将它们保留原样。

DP 配置总结 : PD 控制器使用两组与两个主要 DP 交替模式配置 (4 通道与 2 通道) 相对应的事件, 并分别处理两种方向。在每种情况下, 这都会将新值写入寄存器 0x0A 以切换 TUSB1044 模式 (DP 或 DP + USB, 并设置翻转状态), 并调整 EQ 寄存器 0x10/0x11 (有时还会调整 0x20/0x21) 以在新模式下保持信号完整性。选择的值 (为寄存器 0x0A 选择的 0x1A/1E、0x1B/1F 以及各种 EQ 代码) 源自 TUSB1044 的要求 :

- **0x0A 值** : 主要在低 3 位方面有所不同。CTLSEL = 2 (仅 DP) ; 对于 DP+USB, = 3 ; 对于方向, FLIP = 0 或 1。位 4 始终保持为 1 (我们始终使用 EQ 覆盖模式)。
- **EQ 值** : 我们在上行侧为 USB3 通道使用 0x6, 为 DP 通道使用 0x5, 在下行侧则通用 0x3。这些特定代码来自 TI 的参考设计或实验室调优, 旨在满足 USB3 Gen2 和 DP HBR2/HBR3 合规性要求。重要的部分不是确切

的数字，而是 PD 控制器可以根据场景的需要进行更改。例如，如果某个特定设计需要为检测到的更长电缆提供不同的 EQ，则固件可能会使用不同的值（尽管我们的示例将每个模式的 EQ 保持不变）。

执行 DisplayPort 配置事件后，TUSB1044 完全处于交替模式配置中：通道会通过转接驱动器将 DisplayPort 从系统的 GPU 传输到 USB-C 连接器，并且（如果在 BDF 中）两个通道继续传输 USB3 数据。PD 控制器主动参与在此结束，系统会发送 DisplayPort AUX 握手等来照常设置监视器。TUSB1044 与协议无关，只需通过高速信号即可；我们只通过 I2C 有效地控制开关和增益。

Record index 19 (0x13)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__DP_CONFIG_ACE_UU
Data Length	2
Target Address Index	1
Orientation Independent	☐
Data	0x1a0a

图 4-11. 寄存器 0x0A 上的 DP 分配 ACE 配置事件

Record index 20 (0x14)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__DP_CONFIG_ACE_UU
Data Length	3
Target Address Index	1
Orientation Independent	☐
Data	0x555510

图 4-12. 寄存器 0x10 上的 DP 分配 ACE 配置事件

Record index 21 (0x15)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__DP_CONFIG_ACE_UD
Data Length	2
Target Address Index	1
Orientation Independent	☐
Data	0x1e0a

图 4-13. 寄存器 0x0A 上的 DP 分配 ACE 配置事件

Record index 22 (0x16)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__DP_CONFIG_ACE_UD
Data Length	3
Target Address Index	1
Orientation Independent	☐
Data	0x555510

图 4-14. 寄存器 0x10 上的 DP 分配 ACE 配置事件

Record index 23 (0x17)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__DP_CONFIG_BDF_UU
Data Length	2
Target Address Index	1
Orientation Independent	☐
Data	0x1b0a

图 4-15. 寄存器 0x0A 上的 DP 分配 BDF 配置事件

Record index 24 (0x18)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__DP_CONFIG_BDF_UU
Data Length	3
Target Address Index	1
Orientation Independent	☐
Data	0x665510

图 4-16. 寄存器 0x10 上的 DP 分配 BDF 配置事件

Record index 25 (0x19)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__DP_CONFIG_BDF_UU
Data Length	3
Target Address Index	1
Orientation Independent	☐
Data	0x333320

图 4-17. 寄存器 0x20 上的 DP 分配 BDF 配置事件

Record index 26 (0x1a)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__DP_CONFIG_BDF_UD
Data Length	2
Target Address Index	1
Orientation Independent	☐
Data	0x1f0a

图 4-18. 寄存器 0x0A 上的 DP 分配 BDF 配置事件

Record index 27 (0x1b)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__DP_CONFIG_BDF_UD
Data Length	3
Target Address Index	1
Orientation Independent	☐
Data	0x556610

图 4-19. 寄存器 0x10 上的 DP 分配 BDF 配置事件

Record index 28 (0x1c)	
Field	Value
Trigger Event	I2C_MASTER_EVENT__DP_CONFIG_BDF_UD
Data Length	3
Target Address Index	1
Orientation Independent	☐
Data	0x333320

图 4-20. 寄存器 0x20 上的 DP 分配 BDF 配置事件

通过这些连接和 DP 配置事件处理程序，TPS65992S 可确保在以下每个阶段正确配置 TUSB1044：初始连接（用于 USB 3.2 SuperSpeed 或基线运行）和转换到 DisplayPort 交替模式期间。所有必要的 I2C 写入配置完成后，PD 控制器固件会自动执行，从而无需专用 EC 或 MCU 来实时处理转接驱动器设置。

5 I2C 事件表汇总

通过使用 TPS65992 I2C 控制器功能，我们实现了对电源 (TPS55288) 和信号转接驱动器 (TUSB1044) 的协调控制以响应 Type-C 事件。下面的表 5-1 汇总了所有配置的 I2C 事件，包括索引、触发器、目标器件寄存器和写入的数据：

表 5-1. 所有 I2C 索引汇总

	触发事件	register	value
Index1	I2C_MASTER_EVENT_POWER_ON_RESET	0x00	0xD2
Index2	I2C_MASTER_EVENT_POWER_ON_RESET	0x04	0x03
Index3	I2C_MASTER_EVENT_POWER_ON_RESET	0x06	0xa0
Index4	I2C_MASTER_EVENT_SRC_PDO1_NEGOTIATED	0x00	0xD2 0x00
Index5	I2C_MASTER_EVENT_SRC_PDO2_NEGOTIATED	0x00	0x9A 0x01
Index6	I2C_MASTER_EVENT_SRC_PDO3_NEGOTIATED	0x00	0xC5 0x02
Index7	I2C_MASTER_EVENT_SRC_PDO4_NEGOTIATED	0x00	0xBF 0x03
Index8	I2C_MASTER_EVENT_DETACH	0x00	0xD2
Index9	I2C_MASTER_EVENT_POWER_ON_RESET	0x0A	0x10
Index10	I2C_MASTER_EVENT_DETACH	0x0A	0x10
Index11	I2C_MASTER_EVENT_ATTACH_UU	0x0A	0x11
Index12	I2C_MASTER_EVENT_ATTACH_UU	0x0C	0x58
Index13	I2C_MASTER_EVENT_ATTACH_UU	0x10	0x66 0x66
Index14	I2C_MASTER_EVENT_ATTACH_UU	0x20	0x33 0x33
Index15	I2C_MASTER_EVENT_ATTACH_UD	0x0A	0x15
Index16	I2C_MASTER_EVENT_ATTACH_UD	0x0C	0x58
Index17	I2C_MASTER_EVENT_ATTACH_UD	0x10	0x66 0x66
Index18	I2C_MASTER_EVENT_ATTACH_UD	0x20	0x33 0x33
Index19	I2C_MASTER_EVENT_DP_CONFIG_ACE_UU	0x0A	0x1A
Index20	I2C_MASTER_EVENT_DP_CONFIG_ACE_UU	0x10	0x55 0x55
Index21	I2C_MASTER_EVENT_DP_CONFIG_ACE_UD	0x0A	0x1E
Index22	I2C_MASTER_EVENT_DP_CONFIG_ACE_UD	0x10	0x55 0x55
Index23	I2C_MASTER_EVENT_DP_CONFIG_BDF_UU	0x0A	0x1B
Index24	I2C_MASTER_EVENT_DP_CONFIG_BDF_UU	0x10	0x66 0x55
Index25	I2C_MASTER_EVENT_DP_CONFIG_BDF_UU	0x20	0x33 0x33
Index26	I2C_MASTER_EVENT_DP_CONFIG_BDF_UD	0x0A	0x1F
Index27	I2C_MASTER_EVENT_DP_CONFIG_BDF_UD	0x10	0x55 0x66
Index28	I2C_MASTER_EVENT_DP_CONFIG_BDF_UD	0x20	0x33 0x33

(表格条目反映了上述配置。具有两个数据字节的“REG0x00”表示从 0x00 开始进行多字节写入。如果需要，未使用的索引 8 将保留用于将来的扩展或其他事件。为确保完整性，与方向无关的写入 (如 0x0C、0x10、0x20) 在我们表中的 UU 和 UD 触发条件下都是重复的，但数据是相同的。)

如表中所示，TPS65992S 可以处理各种事件，从端口初始上电到任一方向的电缆连接，再到 USB PD 合约协商和 DisplayPort 交替模式，所有这些事件都是通过执行预定义的 I2C 事务来实现。这种事件驱动的控制方案可保持外部硬件 (电源转换器和转接驱动器) 在每个阶段都处于正确的状态：

- 复位和断开时，电源安全 (5V 或关闭)，高速开关复位。

- 在连接时，电源会保持在 5V，直到请求更高的电压，而 USB3 转接驱动器可正确连接 SuperSpeed 通道。
- 当请求更高电压的 PDO 时，PD 控制器会将 TPS55288 的输出提高到所需的电平。
- 如果进入 DisplayPort 模式，PD 控制器会重新配置转接驱动器，使正确的通道承载 DisplayPort，而任何其余通道承载 USB3，从而根据需要调整 EQ 设置以实现信号完整性。
- 一旦通过定制工具对 PD 控制器进行了编程，所有这些都会自主进行，无需实时软件干预。

结语

借助 TPS65992S 集成式 I2C 主机功能，USB-C PD 系统可以紧密协调电力输送和高速信号切换。在此应用中，TPS65992S 将 TPS55288 降压/升压转换器配置成为每个 PD 合约输出适当的电压，并管理 TUSB1044 转接驱动器以针对任何电缆方向和模式正确路由 USB3/DisplayPort 信号。通过在 PD 控制器固件中仔细对事件排序，可消除逻辑不一致（例如，仅在设置正确的电压后验证转换器是否已启用，并且在模式切换期间转接驱动器绝不会处于未定义状态）。

这种方法将取代原本需要外部 MCU 或复杂 GPIO 逻辑的情况，并采用固件驱动的设计，使用 TI 的工具更易于维护和更新这种设计。每个关键事件（连接、分离、合约协商、模式进入）都会触发一个预定义的序列，从而保持硬件状态始终与 USB-C 状态一致。

用户将获得稳健可靠的设计，例如，当笔记本电脑插入显示器时：Type-C 控制器会立即配置 TUSB1044 以保持正确的方向，然后协商更高的电压并向 TPS55288 发送信号以斜升至 20V 进行充电，而在启动 DisplayPort 交替模式时，会重新配置转接驱动器以使显示器正常运作，这一切只需几秒钟即可完成，无需用户干预。这展示了集成 PD 控制器设计的强大功能。

6 参考资料

- 具有 I2C 接口的 TPS55288 36V、16A 降压/升压转换器
- [TUSB1044 USB TYPE-C™ 10Gbps 多协议双向线性转接驱动器](#)

重要通知和免责声明

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 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。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
版权所有 © 2025，德州仪器 (TI) 公司