

I²C 总线及系统应用指南

接口 AE 团队

摘要

本应用指南系统性地介绍了 I²C（Inter-Integrated Circuit）协议以及 I²C 系统中常见的器件应用。内容涵盖 I²C 物理层、协议基础、速率模式和电气时序规范，并阐述了三类 I²C 系统常用器件—I²C 开关（I²C Switch）、I/O 扩展（I²C to I/O Expander）以及 I²C 电平转换器（I²C Level Translator）的架构原理、功能特性和应用注意事项。以 3PEAK 相关器件为案例，结合实践中的常见问题与处理建议，为工程师提供从协议理解到器件选型、器件工作原理、硬件设计参考的全链路实用指南，助力高效完成 I²C 系统设计。

目录

1.概述.....	7
1.1 I ² C 系统简介.....	7
1.2 I ² C 总线特性.....	8
1.3 I ² C 系统术语表.....	9
2.I ² C 物理层.....	11
2.1 概述.....	11
2.2 开漏输出结构.....	11
2.2.1 开漏输出.....	11
2.2.2 开漏输出与推挽输出对比.....	12
2.3 高电平和上升沿.....	13
2.4 低电平和下降沿.....	14
2.5 I ² C 上拉电阻选取.....	17
3.I ² C 协议.....	20
3.1 I ² C 帧结构.....	20
3.1.1 START 和 STOP.....	20
3.1.2 地址字节和数据字节.....	21
3.1.3 ACK 和 NACK.....	22
3.2 I ² C 读写.....	23
3.2.1 I ² C 读写帧格式.....	23
3.2.2 连续读写与地址自动递增.....	24

3.2.3 I ² C 读写案例.....	24
3.3 I ² C 地址.....	26
3.3.1 I ² C 器件地址类型.....	26
3.3.2 10-Bit 地址.....	27
3.3.3 I ² C 地址冲突.....	28
3.4 I ² C 可选特性.....	29
3.4.1 时钟同步.....	29
3.4.2 仲裁.....	30
3.4.3 时钟延展.....	30
3.4.4 广播地址.....	31
3.4.5 软件复位.....	31
3.4.6 START 字节.....	32
3.5 I ² C 总线挂死和恢复.....	33
4.I ² C 速率模式.....	34
5.I ² C 电气和时序规范.....	36
5.1 I ² C 电气规范.....	36
5.2 I ² C 时序规范.....	37
5.3 高速模式和超快速模式规范.....	39
6.I ² C 开关.....	40
6.1 概述.....	40
6.2 3PEAK I ² C 开关系列.....	40
6.3 I ² C 开关工作原理.....	41
6.4 器件寻址与 I ² C 读写.....	43
6.5 2:1 I ² C 开关 (仲裁器)	44
6.6 I ² C 开关与模拟开关对比.....	44
6.7 应用注意事项.....	45
7.I/O 拓展.....	46
7.1 概述.....	46
7.2 3PEAK 选型.....	46
7.3 特性介绍.....	46
7.4 寄存器描述与 I ² C 读写.....	47
7.5 IO 拓展芯片 I ² C 读写示例.....	49
7.6 应用注意事项.....	50
8.I ² C 电平转换.....	51
8.1 概述.....	51
8.2 3PEAK I ² C 电平转换选型.....	51

8.3 分立 MOS 电平转换电路.....	53
8.3.1 电路结构和工作原理.....	53
8.3.2 优势与劣势.....	53
8.4 集成 MOS 架构电平转换器.....	54
8.4.1 概述.....	54
8.4.2 工作原理.....	54
8.4.3 典型应用电路.....	56
8.4.4 EN 和 V _{REF2} 的常见错误接法.....	57
8.4.5 功耗.....	58
8.4.6 上拉电阻.....	59
8.4.7 推挽电平转换.....	60
8.4.8 集成 MOS 器件相比分立方案 MOS 优势.....	61
8.4.9 应用注意事项.....	62
8.5 带边沿加速的 MOS 架构电平转换器.....	64
8.5.1 概述.....	64
8.5.2 工作原理.....	64
8.5.3 应用注意事项.....	65
8.6 I ² C 中继器.....	66
8.6.1 概述.....	66
8.6.2 工作机制.....	66
8.6.3 TPT29617A 典型 I ² C 应用波形.....	68
8.6.4 应用注意事项.....	69
8.7 热插拔缓冲器.....	71
8.7.1 概述.....	71
8.7.2 工作机制.....	71
8.7.3 热插拔应用设计.....	72
8.7.4 背板分立热插拔方案.....	74
8.7.5 应用注意事项.....	75
9. 修订记录.....	76
10. 参考资料.....	77
11. 声明与提示.....	78

表目录

表 1-1 I ² C 总线特性.....	8
表 1-2 I ² C 系统术语表.....	9
表 3-1 I ² C 总线强制与可选特性.....	29

表 5-1 SDA 和 SCL 的 IO 引脚规范.....	36
表 5-2 I ² C 时序规范.....	37
表 7-1 TPT29555A 寄存器.....	48
表 7-2 TPT29554A 寄存器.....	48
表 8-1 3PEAK I ² C/I ³ C 电平转换、缓冲器.....	51
表 8-2 3PEAK 通用 OD 应用电平转换.....	52
表 8-3 上拉电阻最小值限制.....	59
表 8-4 TPT29617A V _{OL} 和 V _{IL} 特性.....	66

图目录

图 1-1 I ² C 总线.....	7
图 1-2 典型 I ² C 系统案例.....	8
图 2-1 I ² C 总线示意图.....	11
图 2-2 开漏输出结构.....	12
图 2-3 开漏输出非破坏性竞争.....	12
图 2-4 推挽输出破坏性竞争.....	13
图 2-5 上升沿示意图.....	13
图 2-6 下降沿示意图.....	14
图 2-7 V _{OL} 抬升.....	15
图 2-8 存在多个上拉电阻时的 V _{OL}	16
图 2-9 存在 9306, 9548 或分立 MOS 的 V _{OL}	16
图 2-10 存在串阻时的 V _{OL}	17
图 2-11 上拉电阻最大值曲线.....	18
图 2-12 上拉电阻最小值曲线.....	19
图 3-1 I ² C 帧结构.....	20
图 3-2 START 和 STOP.....	21
图 3-3 地址字节.....	21
图 3-4 数据字节.....	21
图 3-5 数据位传输.....	22
图 3-6 ACK 和 NACK.....	22
图 3-7 写操作帧格式.....	23
图 3-8 读操作帧格式.....	23
图 3-9 先写后读操作帧格式.....	24
图 3-10 TPT29548L 写命令.....	25
图 3-11 TPT29548L 读命令.....	25
图 3-12 TPT29508 写命令.....	25
图 3-13 TPT29508 读命令 增加图例.....	25
图 3-14 10-Bit 地址写操作.....	27

图 3-15 10-Bit 地址读操作.....	28
图 3-16 时钟同步.....	30
图 3-17 仲裁.....	30
图 3-18 广播地址.....	31
图 3-19 软件复位命令.....	32
图 3-20 写操作时 ACK 期间挂死和恢复.....	33
图 3-21 读操作时目标设备返回数据期间挂死和恢复.....	33
图 4-1 高速模式 SCL 物理层.....	35
图 4-2 高速模式帧.....	35
图 5-1 I ² C 时序定义.....	38
图 6-1 I ² C 开关结构框图.....	41
图 6-2 I ² C 开关 PassFET 架构.....	41
图 6-3 TPT29548L 控制寄存器.....	42
图 6-4 I ² C 写命令.....	44
图 6-5 I ² C 读命令.....	44
图 6-6 TPT29641 双主机仲裁通讯过程.....	44
图 6-7 I ² C 开关上拉电阻总电流.....	45
图 7-1 TPT29555A 地址.....	47
图 7-2 TPT29555A 单次写入.....	49
图 7-3 TPT29555A 连续写入.....	49
图 7-4 TPT29555A 单次读取.....	49
图 7-5 TPT29555A 连续读取.....	50
图 8-1 分立 MOS 电平转换电路.....	53
图 8-2 TPT29306 集成 MOS 电平转换架构.....	54
图 8-3 NMOS 内部结构图.....	55
图 8-4 分立 NMOS 器件单向体二极管.....	55
图 8-5 集成 MOS 器电平转换器件双二极管.....	55
图 8-6 TPT29306 典型应用电路.....	56
图 8-7 TPT29306 推挽信号使能控制电路.....	56
图 8-8 TPT29306 开漏信号使能控制电路.....	57
图 8-9 错误接法：EN 直接接高压侧电源 V _{CC2}	57
图 8-10 错误接法：EN 直接接推挽信号.....	58
图 8-11 错误接法：EN 电阻上拉，但没有与 V _{REF2} 短接.....	58
图 8-12 功耗.....	59
图 8-13 推挽电平转换.....	60
图 8-14 高压侧向低压侧推挽电平转换波形.....	61
图 8-15 低压侧向高压侧推挽电平转换波形.....	61
图 8-16 TPXF0204 结构框图.....	62

图 8-17 集成边沿加速功能的 MOS 电平转换芯片架构示意图.....	64
图 8-18 开漏信号波形.....	65
图 8-19 推挽信号波形.....	65
图 8-20 锁死机制.....	66
图 8-21 TPT29617A 1 侧向 2 侧电平传输过程.....	67
图 8-22 TPT29617A 1 侧向 2 侧电平传输波形.....	67
图 8-23 TPT29617A 2 侧向 1 侧电平传输过程.....	68
图 8-24 TPT29617A 2 侧向 1 侧电平传输波形.....	68
图 8-25 1 侧为控制器，2 侧为目标器件时总线波形.....	68
图 8-26 2 侧为控制器，1 侧为目标器件时总线波形.....	69
图 8-27 TPT29617A 正确级联接法 1.....	70
图 8-28 TPT29617A 正确级联接法 2.....	70
图 8-29 TPT29617A 错误级联接法.....	70
图 8-30 TPT29511 上升时间加速波形.....	72
图 8-31 TPT29511 热插拔应用.....	72
图 8-32 板卡金手指设计.....	72
图 8-33 背板分立热插拔方案.....	74
图 8-34 使用 TPT29545 的背板分立热插拔方案.....	75

1. 概述

1.1 I²C 系统简介

I²C（Inter-Integrated Circuit）是一种简单、高效的双向同步串行通信总线，由飞利浦公司（现恩智浦半导体）于 20 世纪 80 年代开发。它仅通过两根信号线：串行数据线（SDA）和串行时钟线（SCL），即可实现设备间的数据交换，极大简化了硬件连接与系统设计。I²C 总线支持多主从架构，允许多个主设备与多个从设备在同一总线上通信，每个从设备拥有唯一的地址。由于其结构简洁、占用引脚少、成本低廉，被广泛应用于连接微控制器与各种低速外设，如传感器、存储器、IO 扩展芯片等，成为嵌入式系统中不可或缺的板级通信标准之一。并以其为基础衍生出了用于主板设备管理的 SMBus（系统管理总线）和专用于电源管理的 PMBus（电源管理总线）等增强型协议。

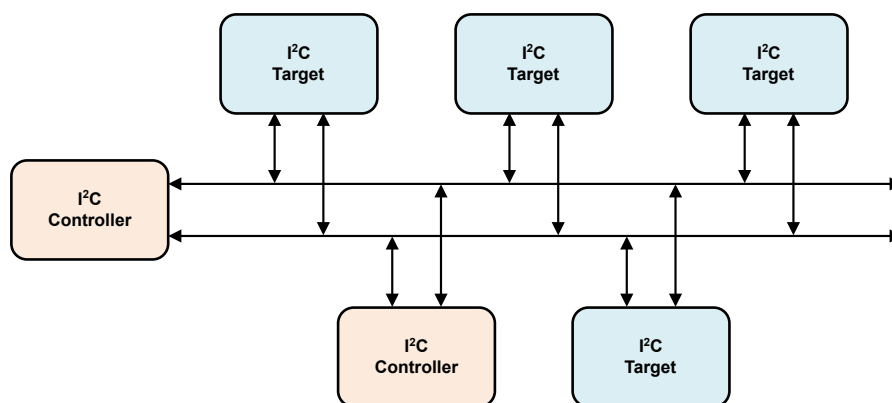


图 1-1 I²C 总线

I²C 系统通常由主控制器（主机）、目标器件（从机）、电平转换器件、总线扩展器件等设备构成，如图 1-2 所示。主控制器一般为微控制器（MCU）或片上系统（SoC），负责发起总线传输并产生时钟信号。目标器件涵盖各类 I²C 接口外设，如温度传感器、模数/数模转换器（ADC/DAC）、EEPROM 存储器以及 I/O 扩展芯片等，各器件通过唯一地址在同一总线上区分。

随着系统规模扩大，单一总线段常面临地址冲突与负载电容超限的问题。当多个同型号器件（地址完全相同）需接入同一系统时，可通过 I²C 开关（I²C Switch）将总线分为多个下游通道，各器件置于不同通道，由主控分时选通访问。当总线负载电容较重或驱动距离过长时，可引入 I²C 中继器（I²C Repeater）或 I²C 缓冲器（I²C Buffer）将总线电容分段隔离，并重驱动信号来满足总线速率和时序要求。在多电压域 I²C 系统中，不同器件的总线电压可能不一致，此时需要使用 I²C 电平转换器件实现跨电压域双向通信。通过 I²C 开关、缓冲器与电平转换器件的組合使用，系统设计者可将原本单段、单电压的 I²C 总线扩展为多段、多电压域的总线拓扑，实现复杂板级互联需求。

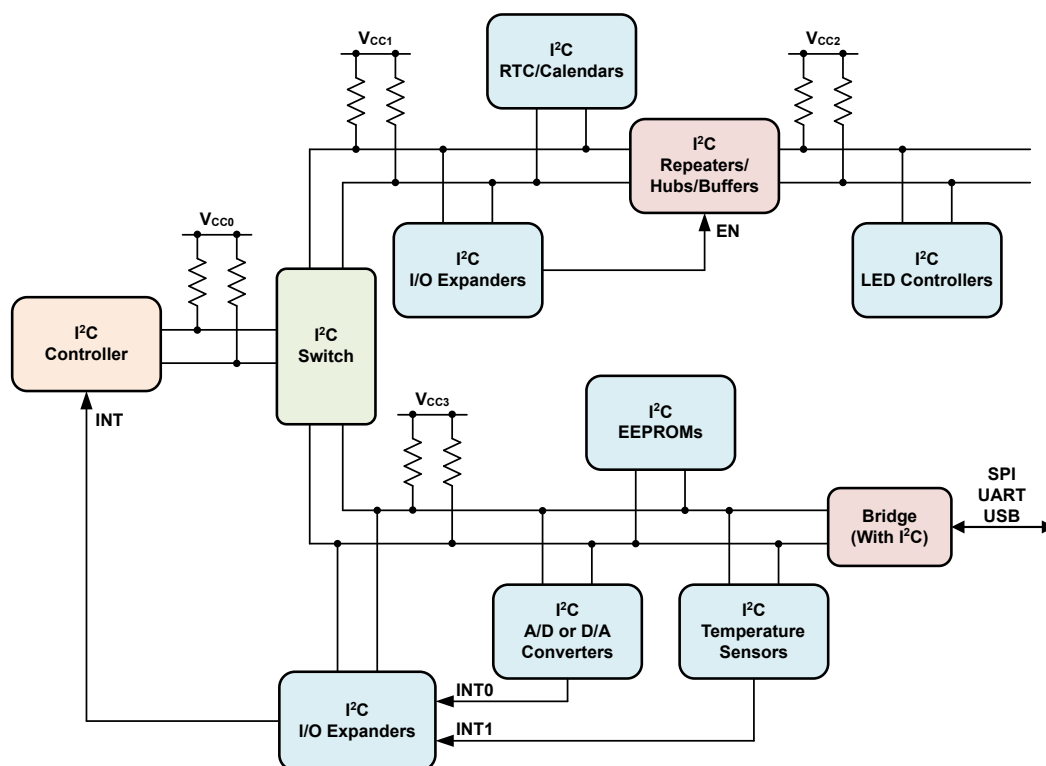


图 1-2 典型 I²C 系统案例

1.2 I²C 总线特性

表 1-1 I²C 总线特性

特性	规格描述
总线线路	串行数据线（SDA）+ 串行时钟线（SCL）
通信方式	同步、串行、半双工
输出类型	开漏输出，需外加上拉电阻
最高速率	标准模式（Standard-mode）：100 kbps 快速模式（Fast-mode）：400 kbps 快速增强模式（Fast-mode Plus）：1 Mbps 高速模式（High-speed mode）：3.4 Mbps 超快速模式（Ultra Fast-mode）：5 Mbps
拓扑结构	多主多从
设备寻址	软件寻址，每个设备具有唯一地址
仲裁机制	支持总线仲裁
应答机制	支持

特性	规格描述
抗干扰能力	集成 50ns 滤波器抑制毛刺干扰
扩展性	理论寻址模式的最大节点数为 2^7 (7-Bit 地址) 或 2^{10} (10-Bit 地址)，但有保留 16 个地址用于特殊用途。实际连接设备数量受总线最大电容限制

I²C 总线的核心特性如表 1-1 所示，详细协议与特性说明参见后续章节。

1.3 I²C 系统术语表

I²C 系统常见术语及解释如下表。

表 1-2 I²C 系统术语表

术语	描述
I ² C	Inter-Integrated Circuit，通常简写为 I ² C。
SDA	I ² C 串行数据线 (Serial Data Line)
SCL	I ² C 串行时钟线 (Serial Clock Line)
控制器/主机 (Controller/Master)	发起总线传输、产生时钟信号和终止传输的设备。在最新的 I ² C/I ³ C 协议规范中，“主机”相关表述已由“Master”更新为“Controller”。本文中“控制器”与“主机”指代同一概念。
目标设备/从机 (Target/Slave)	被控制器寻址并响应其命令的设备。在最新的 I ² C/I ³ C 协议规范中，“从机”相关表述已由“Slave”更新为“Target”。本文中“目标设备”与“从机”指代同一概念。
上拉电阻	连接 SDA/SCL 至总线电源 V _{CC} 的电阻，为开漏总线提供高电平。
Start	I ² C 通讯帧起始条件，在总线空闲状态下发起传输，通常简写为“S”。
Stop	I ² C 通讯帧停止条件，通常简写为“P”。
Repeated Start	重复起始条件，在控制器不先发出停止条件的情况下，于一次总线传输期间再次发出起始条件。重复起始条件用于在不释放总线控制权的前提下发起新的通讯。通常简写为“Sr”。
仲裁 (Arbitration)	I ² C 总线多控制器同时尝试控制总线时，确保仅一个控制器获得控制权且其数据不被破坏的机制。
时钟同步 (Synchronization)	I ² C 总线两个或多个设备的时钟信号同步的机制。
时钟延展 (Clock Stretching)	目标设备拉低 SCL 以暂停总线传输，待就绪后释放 SCL 继续通信。
线与 (Wired-AND)	开漏输出配合上拉电阻实现的逻辑“与”功能，任一设备拉低总线则总线为低

术语	描述
应答 (ACK/NACK)	接收器在每字节传输后发出的确认 (ACK) 或非确认 (NACK) 位。
I ² C 开关 (I ² C Switch)	通过 I ² C 命令控制多路下游通道选通的器件，用于总线拓扑扩展与解决地址冲突。
I ² C 中继器 (I ² C Repeater)	具备信号重驱动和两侧电容隔离功能的器件，用于延长总线距离或分段隔离负载。
I ² C 缓冲器 (I ² C Buffer)	同中继器，部分缓冲器额外支持热插拔保护功能。
I ² C I/O 扩展芯片 (I ² C I/O Expander)	通过 I ² C 总线拓展额外 GPIO 端口的器件，各端口可独立配置为输入或输出。
I ² C 电平转换器 (I ² C Level Translator)	在两个不同电压的总线之间实现双向 I ² C 电平转换的器件。

2. I²C 物理层

2.1 概述

I²C 系统为总线上的所有设备提供了两条共享的通信线路：**SCL** 和 **SDA**。其中 **SCL** 是串行时钟线，用于传输由控制器（**Controller/Master**）生成的时钟。**SDA** 是串行数据线，用于向目标设备（**Target/Slave**）发送数据或从目标设备接收数据，实现双向、半双工数据传输。I²C 支持多个控制器和多个目标设备，所有设备同时挂在总线上，通过唯一的软件地址进行寻址。I²C 器件均为开漏（**Open-Drain**）输出或开集电极（**Open-Collector**）输出结构，总线必须通过电阻或者电流源上拉到电源电压（ V_{CC} ）。这种架构可以实现“线与”（**Wired-AND**）逻辑，使得任何设备都可以安全地将总线拉低而不会造成破坏性的总线竞争，在此基础上可以实现数据双向传输，地址仲裁，时钟延展等功能。

同一总线上的所有器件需要公共接地（**GND**）。逻辑电平以电源电压（ V_{CC} ）为参考。一般情况下，低电平输入（ V_{IL} ）电压阈值为 $30\%V_{CC}$ ，高电平输入（ V_{IH} ）电压阈值为 $70\%V_{CC}$ 。

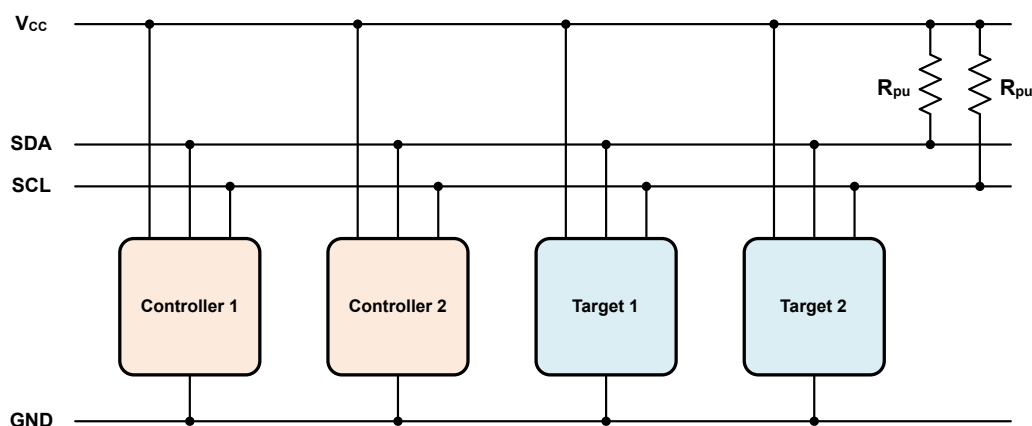


图 2-1 I²C 总线示意图

2.2 开漏输出结构

2.2.1 开漏输出

I²C 器件采用开漏（或开集）输出结构，其输出由一个源极接地的 **NMOS** 构成，如图 2-2 所示。当需要输出逻辑低电平逻辑“0”时，**MOS** 导通，将总线电平下拉至接近地电位；当需要输出高电平逻辑“1”时，**MOS** 关断，总线依靠外部上拉电阻被拉至电源电压（ V_{CC} ）。I²C 总线处于空闲状态时，通过电阻上拉保持高电平。

I²C 开漏的总线模式可以实现“线与”功能，总线输出为所有设备输出的逻辑“与”后的结果。多个设备可以连接在一起而不会出现破坏性争用，确保了多设备共享总线的可靠运行。如果任意一个设备输出逻辑“0”，总线将被拉至低电平。仅当所有设备均输出高电平（即释放总线，呈现高阻态）时，总线才由上拉电阻拉高。此特性确保了低电平具有更高的优先级，是实现多主机仲裁、时钟同步与时钟延展等协议功能的硬件基础。

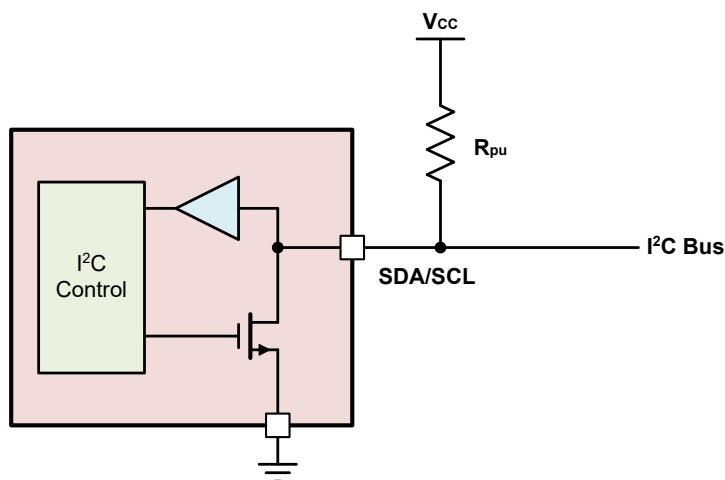


图 2-2 开漏输出结构

2.2.2 开漏输出与推挽输出对比

I²C 开漏输出模式高电平通过电阻上拉实现，任意设备输出低电平时，线路最终都表现为低，且不会对其它想要输出高电平的器件造成破坏性竞争。推挽输出结构一般使用一对互补的 PMOS 和 NMOS 管来作为输出驱动，当 PMOS 打开 NMOS 关闭时，输出高电平；当 PMOS 关闭 NMOS 打开时，输出低电平。作为对比，如果在总线上有两个推挽输出设备处于工作状态且同时驱动相反的电平（一个驱动为高，另一个驱动为低）时，将产生严重的“总线竞争”。一方面，输出电平会稳定在一个不确定的中间值，导致逻辑错误；另一方面，电源（V_{CC}）与地（GND）之间由一个设备的 PMOS 和另一个设备的 NMOS 形成低阻抗通路，产生大电流，可能损坏器件。因此，推挽输出适用于点对点或由单一主机完全控制的单向通信场景。

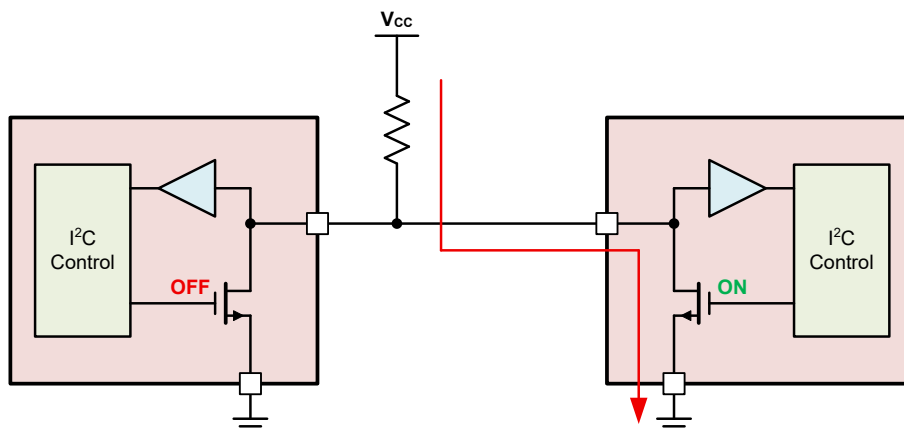


图 2-3 开漏输出非破坏性竞争

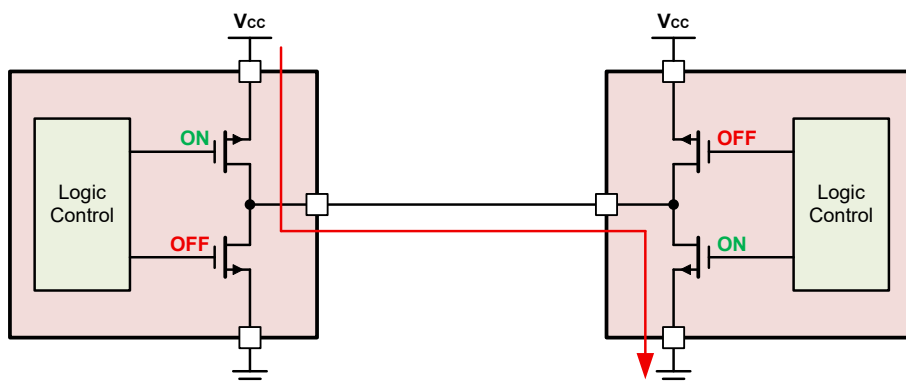


图 2-4 推挽输出破坏性竞争

开漏输出模式下，高电平依赖外部上拉电阻建立，速率受到总线电容和上拉电阻大小限制。推挽输出通过 PMOS 上拉，驱动能力强，可支持更高的通信速率，例如 SPI 通讯、I²C 超快速模式（Ultra Fast-mode）、I³C 通讯。在 I²C 超快速模式规范中，器件输出为推挽驱动，最大速率可以提升至 5 Mbit/s。但推挽输出不具备线与功能，因此该模式仅支持控制器向目标器件的单向写入，不再支持读操作、ACK 应答及多主仲裁等机制。在 I³C 协议中，SCL 固定为推挽输出，而 SDA 可以根据协议自动切换输出模式：地址帧期间 SDA 工作于开漏模式，可以支持地址仲裁，ACK 应答；ACK 后的数据帧期间 SDA 则切换至推挽输出。I³C SDR 通信速率可以达到 12.5 Mbit/s。

2.3 高电平和上升沿

I²C 总线的高电平及上升沿是由上拉电阻驱动。信号从低到高的上升沿波形为 RC 充电波形，其上升时间（ t_{RISE} ）取决于上拉电阻阻值（ R_{PU} ）与总线总寄生电容（ C_b ）。与推挽输出相比，此上升沿相对缓慢，其速率直接限制了总线所能支持的最高通信频率。同时，总线电容也制约了所能挂接的器件数量与走线长度，I²C 的各模式规范中均对最大总线电容（ C_b ）设有明确限制。

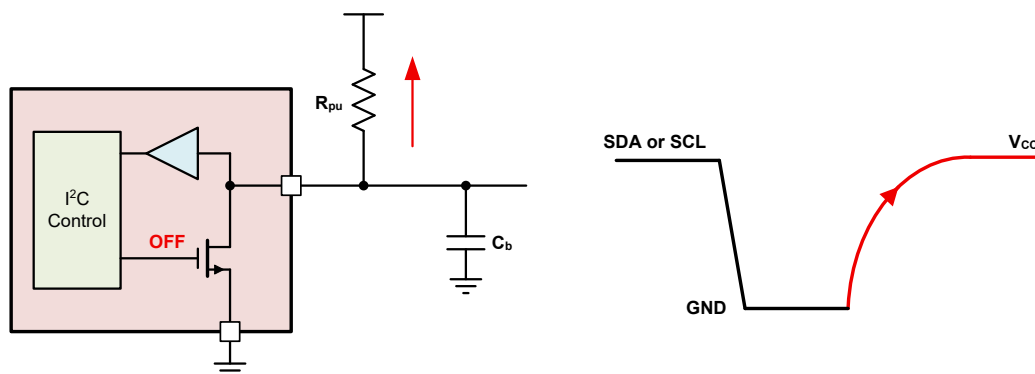


图 2-5 上升沿示意图

I²C 总线上升时间计算

当初始低电平电压为 0V，I²C 信号上升沿期间总线电压 $V(t)$ 随时间 t 变化的公式为：

$$V(t) = V_{\text{CC}} \times \left(1 - e^{-\frac{t}{R_{\text{PU}}C_b}}\right) \quad (1)$$

对于 I²C 电平，其高电平 70%V_{CC} 和低电平 30%V_{CC} 对应的上升时间分别为：

$$V(t_1) = V_{CC} \times \left(1 - e^{-\frac{t_1}{R_{pu}C_b}}\right) = 0.3 \times V_{CC} \quad (2)$$

$$V(t_2) = V_{CC} \times \left(1 - e^{-\frac{t_2}{R_{pu}C_b}}\right) = 0.7 \times V_{CC} \quad (3)$$

计算得到 t₁ 和 t₂：

$$t_1 = R_{pu}C_b \times \ln 0.3 \quad (4)$$

$$t_2 = R_{pu}C_b \times \ln 0.7 \quad (5)$$

I²C 电平上升（30%V_{CC}~70%V_{CC}）时间 t_r 为：

$$t_r = t_2 - t_1 = R_{pu}C_b \times (\ln 0.7 - \ln 0.3) = 0.8473 \times R_{pu}C_b \quad (6)$$

经过上述计算最终得到 I²C 电平上升（30%V_{CC}~70%V_{CC}）时间 t_r 为 0.8473×R_{pu}C_b，应用中需要根据总线电容大小，选取合适的上拉电阻，使得上升时间 t_r 满足协议要求。详细的上拉电阻选取方法如 [I²C 上拉电阻选取](#) 小节所示。

2.4 低电平和下降沿

I²C 总线的低电平由器件内部的 MOS 管主动下拉形成。下降沿是开关管从关断到导通的过程，其速率主要取决于开关管自身的驱动能力和总线负载电容，下降沿通常远快于上升沿。这种不对称的边沿速率是开漏结构的典型特征。

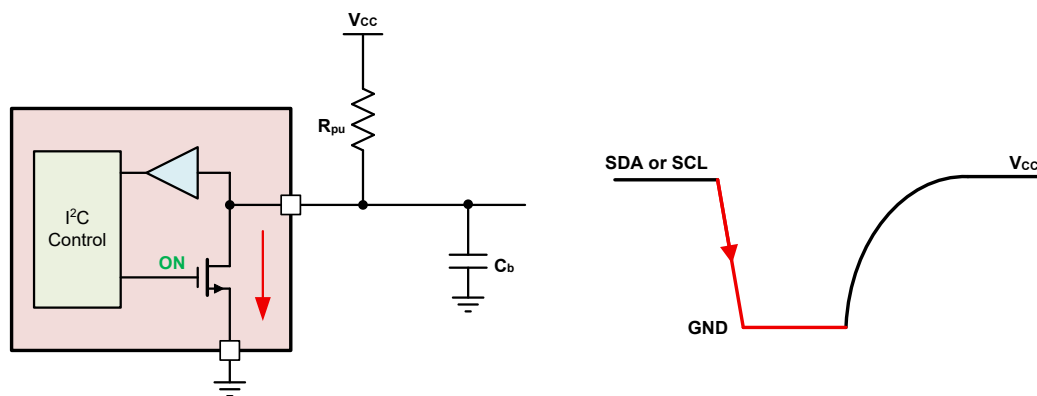


图 2-6 下降沿示意图

快速的下降沿可能带来信号完整性的挑战。当 PCB 走线较长且阻抗不匹配或不连续时，陡峭的下降沿易引起信号反射，形成振铃或过冲。为此，可在驱动器输出端串联一个小电阻以减缓边沿，抑制振铃或过冲。

I²C 总线器件输出低电平（V_{OL}）并非理想的地电位。当下拉 MOS 导通时，下拉电流会流经其导通内阻（R_{on}），从而产生一定的压降，表现为输出低电平的抬升。

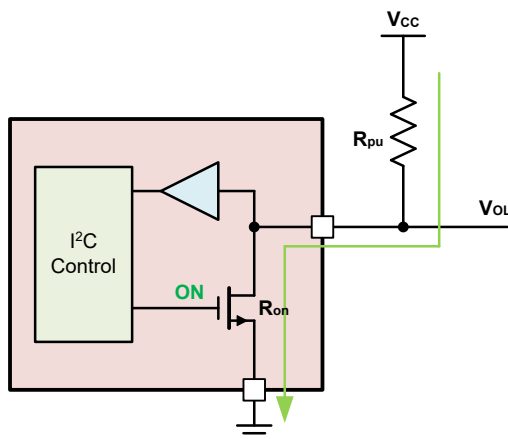


图 2-7 V_{OL} 抬升

低电平抬升公式为：

$$V_{OL} = I_{OL} \times R_{on} = \frac{V_{CC}}{(R_{pu} + R_{on})} \times R_{on} \quad (7)$$

一般情况下 MOS 内阻 R_{on} 远小于上拉电阻 R_{pu} ，因此 V_{OL} 也可以估算为：

$$V_{OL} = I_{OL} \times R_{on} \approx \frac{V_{CC}}{R_{pu}} \times R_{on} \quad (8)$$

其中 I_{OL} 为流过下拉 MOS 的电流总和。

由此可见，器件输出低电平 V_{OL} ，与器件的下拉 MOS 内阻，即器件的驱动能力有关。当 MOS 内阻越大，即低电平驱动能力越弱，低电平抬升越多。I²C 协议对标准模式和快速模式器件低电平驱动能力一般规范如下：3mA 灌电流（ I_{OL} ）下， V_{OL} 低于 0.4V。在器件的数据手册中可以查找到该项参数。有的器件手册中表述为 0.4V VOL 时，IOL 电流大于 3mA，这两种表述相同。

存在多路上拉电阻、串阻、或 MOS 器件时 V_{OL} 计算：

实际应用中，总线上可能有多路上拉电阻，此外总线上可能还有串阻，电平转换器件等，可能额外导致低电平抬升。

a) 总线存在多个上拉电阻

如果总线有多个上拉电阻，则低电平期间所有上拉电阻上都会产生电流。以图 2-8 为例，此时 V_{OL} 计算为：

$$V_{OL} = I_{OL} \times R_{on} = (I_{OL1} + I_{OL2}) \times R_{on} \quad (9)$$

计算总 I_{OL} 时，需要将每路 I_{OL} 累加，或者也可以将电阻换算为所有电阻的并联值进行计算。

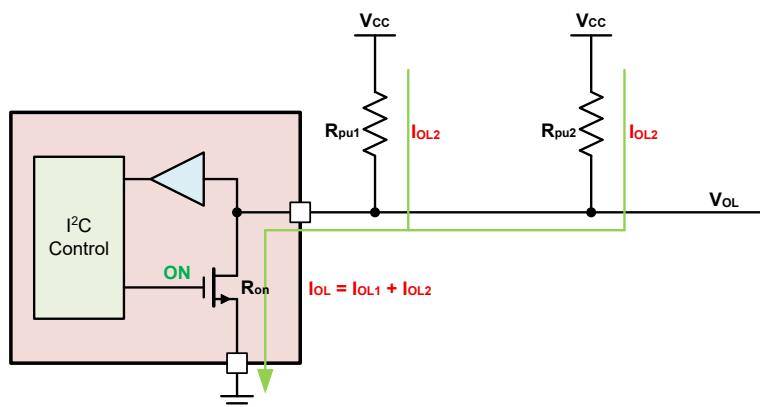


图 2-8 存在多个上拉电阻时的 V_{OL}

b) 总线存在 9306，9548 等 MOS 类型的电平转换器件

如果电路中有 9306，9548 这类 MOS 类型的电平转换，或者存在模拟开关，分立 MOS 器件等情况下，其低电平期间 MOS 导通，因此两侧总线上拉电阻电流也会累加。同时这类电平转换器件的 MOS 本身也存在内阻 R_{on} ，在这种情况下需要考虑各路 I_{OL} 在每级 MOS 上的产生的压降，谨慎选取上拉电阻。

以图 2-9 为例，此时 V_{OL} 计算为：

$$V_{OL} = (I_{OL1} + I_{OL2}) \times R_{on1} + I_{OL2} \times R_{on2} \quad (10)$$

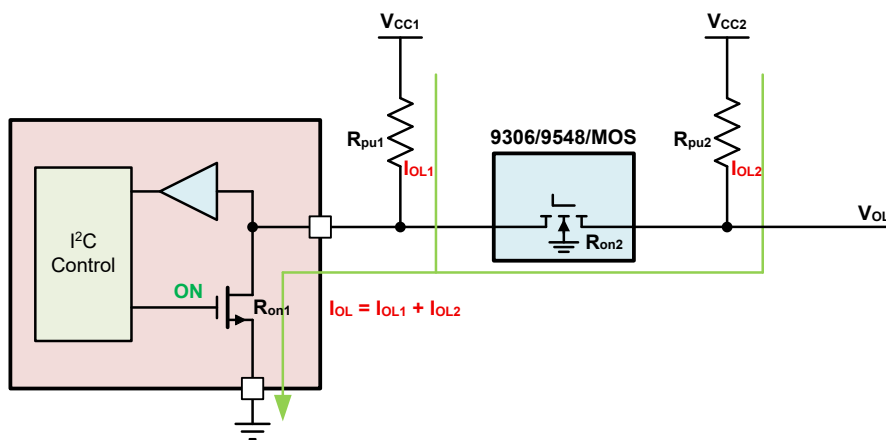


图 2-9 存在 9306，9548 或分立 MOS 的 V_{OL}

c) 总线存在串阻

如果总线存在串阻， I_{OL} 流过串阻也会产生额外压降。以图 2-10 为例，此时 V_{OL} 计算为：

$$V_{OL} = I_{OL} \times (R_{on} + R_s) \quad (11)$$

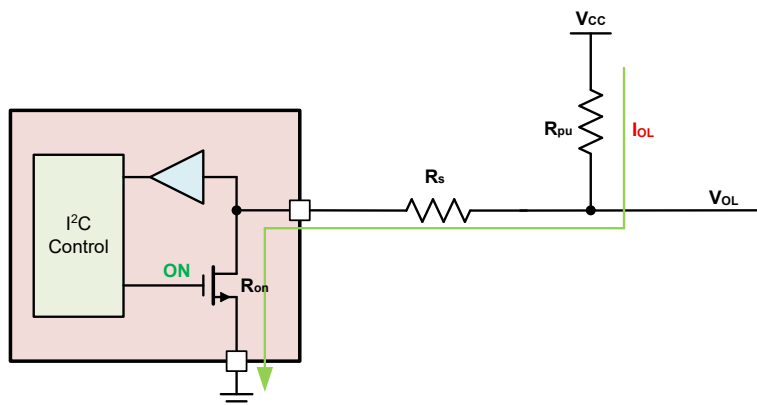


图 2-10 存在串阻时的 V_{OL}

2.5 I²C 上拉电阻选取

上拉电阻的大小，会影响总线上升沿速率和低电平幅度。当上拉电阻选取较大时，上升沿速率会变慢。当上拉电阻选取较小时，则 V_{OL} 抬升幅度变大，同时低电平期间上拉电阻产生的额外功耗也会增大。应用中上拉电阻选取，可以参考如下方法，选取合适的上拉电阻。

a) 上拉电阻最大值

如高电平和上升沿节所述，上升沿速率取决于上拉电阻 R_{pu} 和总线寄生电容 C_b 。

根据公式 6，当上拉电阻应满足：

$$R_{pu} \leq \frac{t_r(\max)}{0.8473 \times C_b} \quad (12)$$

I²C 协议对不同模式的上升时间最大值规定如下：

模式	t_r		单位
	最小值	最大值	
标准模式（100kHz）		1000	ns
快速模式（400kHz）		300	ns
快速增强模式（1MHz）		120	ns

因此，当总线电容和通讯速率确定时，可以计算得到上拉电阻允许的最大值。

标准模式（100kHz）：

$$R_{pu} \leq \frac{1000ns}{0.8473 \times C_b} \quad (13)$$

快速模式（400kHz）：

$$R_{pu} \leq \frac{300ns}{0.8473 \times C_b} \quad (14)$$

快速增强模式（1MHz）：

$$R_{pu} \leq \frac{120ns}{0.8473 \times C_b} \quad (15)$$

将上述公式绘制为曲线：

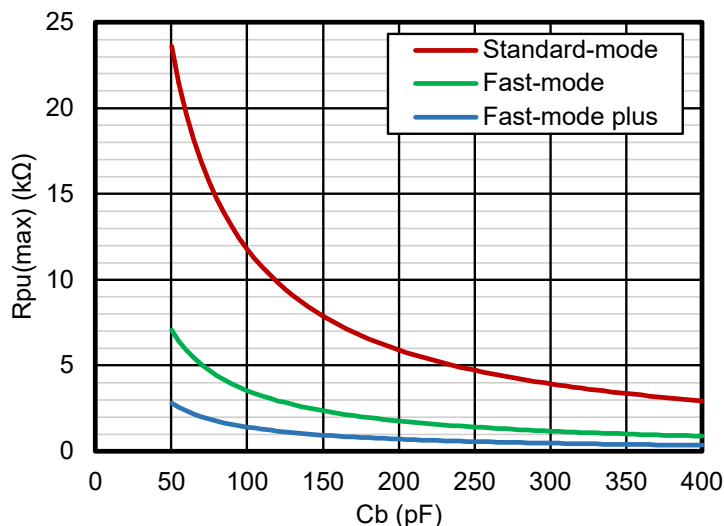


图 2-11 上拉电阻最大值曲线

实际应用中，可通过估算，仿真，实测等方式确定总线寄生电容大小，再计算或查找曲线确定可以选取的上拉电阻最大值。

b) 上拉电阻最小值

上拉电阻最小值主要受总线器件 V_{OL} 驱动能力限制，实际应用中需要查找数据手册，确定器件的 V_{OL} 和 I_{OL} 能力。

为了满足器件低电平要求，上拉电阻应满足：

$$R_{pu} \geq \frac{V_{CC} - V_{OL(max)}}{I_{OL}} \quad (16)$$

I²C 协议对器件的 V_{OL} 有如下规定：

参数	描述	条件	标准模式		快速模式		快速增强模式		单位
			最小值	最大值	最小值	最大值	最小值	最大值	
V_{OL}	低电平输出电压	3 mA 灌电流， $V_{CC} > 2 V$	0	0.4	0	0.4	0	0.4	V
	低电平输出电压	2 mA 灌电流， $V_{CC} \leq 2 V$			0	$0.2V_{CC}$	0	$0.2V_{CC}$	V

因此上拉电阻最小值应满足：

$$R_{pu} \geq \begin{cases} \frac{V_{CC} - 0.4V}{3mA}, & V_{CC} > 2V, \\ \frac{V_{CC} - 0.2 \times V_{CC}}{2mA}, & V_{CC} \leq 2V. \end{cases} \quad (17)$$

绘制为曲线：

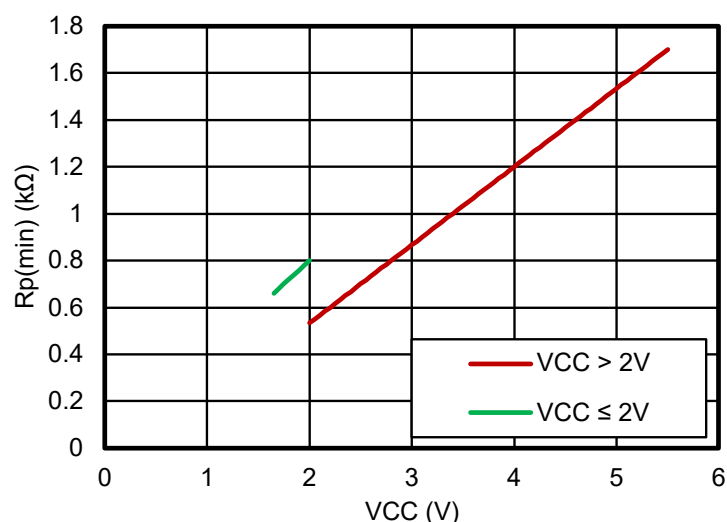


图 2-12 上拉电阻最小值曲线

较小的电阻值对应更小的 RC 时间常数，有助于获得更快的信号边沿，从而支持更高的通信速率；而较大的电阻值能显著降低低电平状态下的静态电流，有利于实现更低的系统功耗。实际选型时，应在满足时序与电平规范的前提下，根据具体应用对速度与功耗的要求进行取舍。

c) 上拉电阻选取案例

以寄生电容 100pF，3.3V I²C 总线为例，进行快速模式（Fast-mode，400kHz）I²C 通信，上拉电阻值过程如下：

$$R_{pu(max)} = \frac{t_r}{0.8473 \times C_b} = \frac{300ns}{0.8473 \times 100pF} = 3.54k\Omega \quad (18)$$

$$R_{pu(min)} = \frac{V_{CC} - V_{OL(max)}}{I_{OL}} = \frac{3.3V - 0.4V}{3mA} = 966.67\Omega \quad (19)$$

因此，上拉电阻可以选择介于 966.67 Ω 和 3.54 kΩ 之间的电阻值。

3. I²C 协议

3.1 I²C 帧结构

I²C 帧由以下部分组成：

- **START/Repeated START**：起始或重复起始条件，作为每次传输的发起信号
- **Addr Byte**：地址字节，用于寻址目标设备，并指明数据传输方向
- **Data Byte**：数据字节：传输的实际信息内容
- **ACK**：应答位，跟随在每个地址或数据字节后，用于确认接收状态
- **STOP**：停止条件：作为每次传输的结束信号

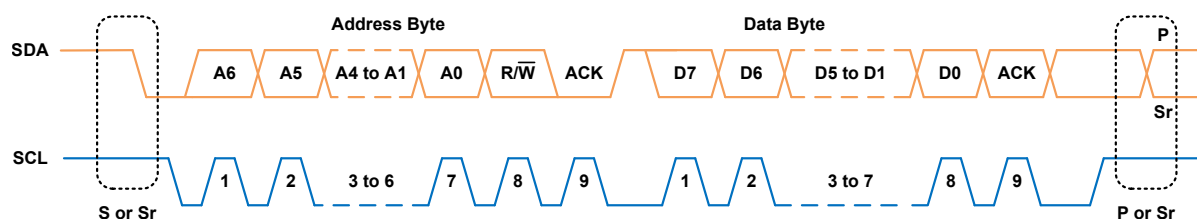


图 3-1 I²C 帧结构

3.1.1 START 和 STOP

I²C 通信由控制器发起，总是以 **START (S)** 或 **Repeated START (Sr)** 开始，并以 **STOP (P)** 条件结束。

a) 起始条件 (**START**)：

当总线处于空闲状态时，控制器通过在 **SCL** 为高电平期间，产生一个 **SDA** 从高到低的下降沿来生成 **START** 条件。**START** 条件标志一次传输的开始，同时该总线被视为处于忙碌状态，总线上其他控制器将暂缓通信。总线上的所有目标设备被唤醒并准备接收后续的地址帧。

b) 停止条件 (**STOP**)：

控制器完成通信后，先释放 **SCL** 至高电平，再释放 **SDA** 产生一个从低到高的上升沿，生成 **STOP** 条件。**STOP** 条件标志一次传输的终止，并释放总线控制权。在 **STOP** 条件出现并经过一段规定的总线空闲时间 (t_{BUF}) 后，总线恢复“空闲”状态，允许其他控制器发起新的通信。

c) 重复起始条件 (**Repeated START**)：

控制器可以在不发出 **STOP** 条件终止当前传输的情况下，再次发出一个 **START** 条件，此即 **Repeated START** 条件。其电平时序与普通 **START** 条件完全相同。**Repeated START** 能在控制器保持总线控制权的前提下，开启一次新的传输序列。常用于：

- (1) 在读写操作间切换方向（例如，先写入寄存器地址，再读取该地址的数据）。
- (2) 在不释放总线的情况下，连续访问多个不同的目标设备。

(3) 重复尝试传输、同步多个 IC 等等。

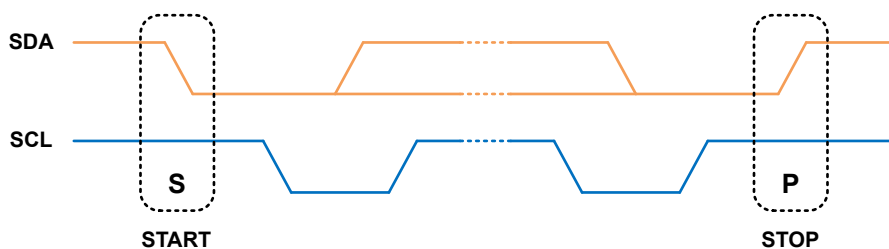


图 3-2 START 和 STOP

3.1.2 地址字节和数据字节

I²C 协议以帧为基本单位进行通信。在控制器发出起始 (S) 或重复起始 (Sr) 条件后，总线进入数据传输阶段，控制器首先发送一个地址字节，随后可发送一个或多个数据字节。

地址字节：

地址字节由 7 位目标设备地址（每个 I²C 目标设备均拥有一个唯一的地址）和 1 位方向位 ($\overline{R/W}$) 构成。7 位地址用于选择总线上特定的目标设备。方向位决定了后续数据的传输方向：当该位为“0”时，表示控制器向目标设备写入数据；为“1”时，表示控制器从目标设备读取数据。

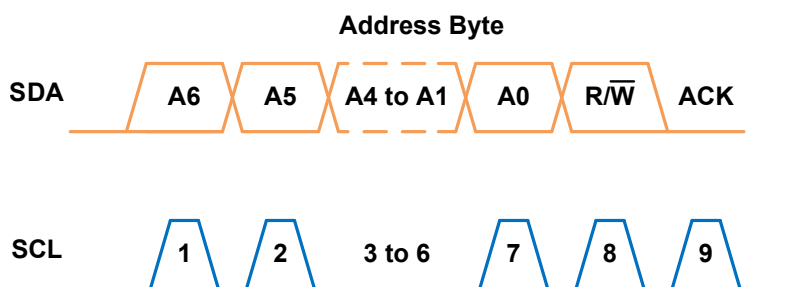


图 3-3 地址字节

数据字节：

数据字节为 8 位，用于传输具体的命令、寄存器地址或寄存器值等信息，其含义完全由目标设备的内部协议定义。地址或数据字节以最高有效位 (MSB) 开始。每个地址或数据字节之后，都跟随一个应答位 (ACK)，见 ACK 章节。

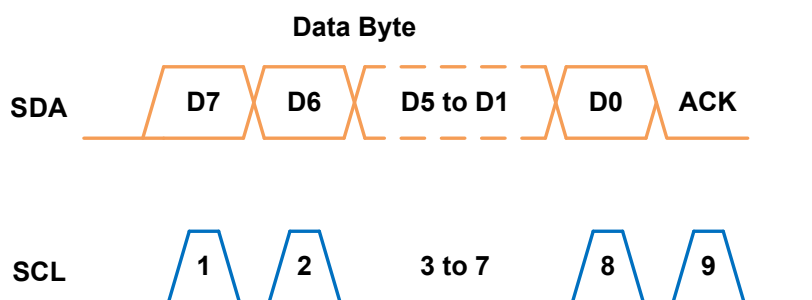


图 3-4 数据字节

数据位传输：

I²C 通信通过 SDA 线传用来输数据位，SCL 线提供同步时钟信号。每个 SDA 数据位对应一个 SCL 时钟脉冲。SDA 的高/低电平状态仅允许在 SCL 时钟的低电平期间发生变化。在 SCL 的高电平期间，SDA 线上的数据必须保持稳定。否则若在此期间 SDA 发生跳变，则可能被解析为 I²C 总线 START 或 STOP 条件。

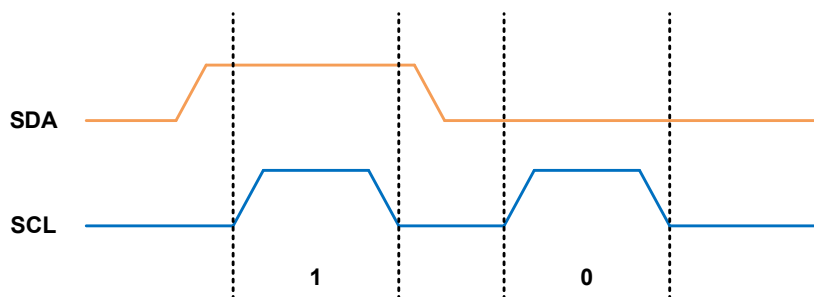


图 3-5 数据位传输

3.1.3 ACK 和 NACK

在 I²C 协议中，每一个传输的地址或数据字节（8 Bits）后都紧跟一个应答位（Acknowledge，简称 ACK）或非应答位（Not Acknowledge，简称 NACK）。该位由接收方在第 9 个 SCL 时钟周期内产生，若接收方将 SDA 线拉低，则表示 ACK，表明字节已成功接收并准备继续；若 SDA 线保持高电平，则表示 NACK，表明字节未成功接收或者地址字节不匹配。

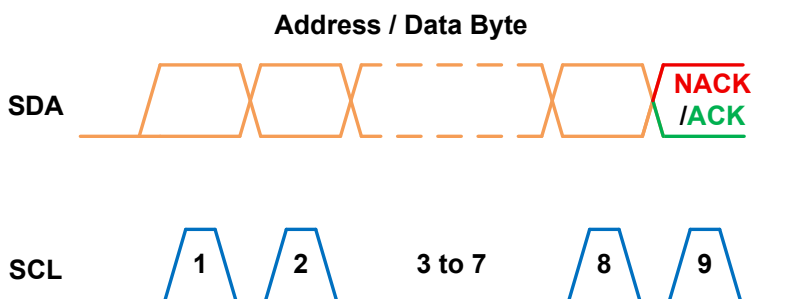


图 3-6 ACK 和 NACK

在地址帧及控制器写数据帧，目标设备为接收方，ACK 由目标设备产生；而在控制器读数据帧后，控制器为接收方，ACK 则由控制器产生。在读取最后一个字节后，控制器必须发送 NACK 以指示传输结束。之后制器可以生成一个 STOP 信号以终止传输，或者生成一个 ReStart 以开始新的传输。

产生 NACK 的常见情况包括：总线上无地址匹配的设备、目标设备忙、接收到无法解析的数据、接收方无法处理更多数据，或控制器主动终止读取。在收到 NACK 后，控制器通常应发出停止条件或重复起始条件以处理通信异常。

3.2 I²C 读写

3.2.1 I²C 读写帧格式

基于基本的帧结构，控制器与目标设备间的 I²C 通信可组合为三种典型序列：写操作、读操作、先写后读操作。

写操作

控制器在发送起始条件（S）和地址帧（R/W=0）后，连续发送一个或多个数据字节。传输方向始终为控制器至目标设备。目标设备在成功接收每个字节（包括地址字节）后，在第 9 个时钟周期内将 SDA 拉低以回复 ACK。传输结束时，控制器产生停止条件（P）。

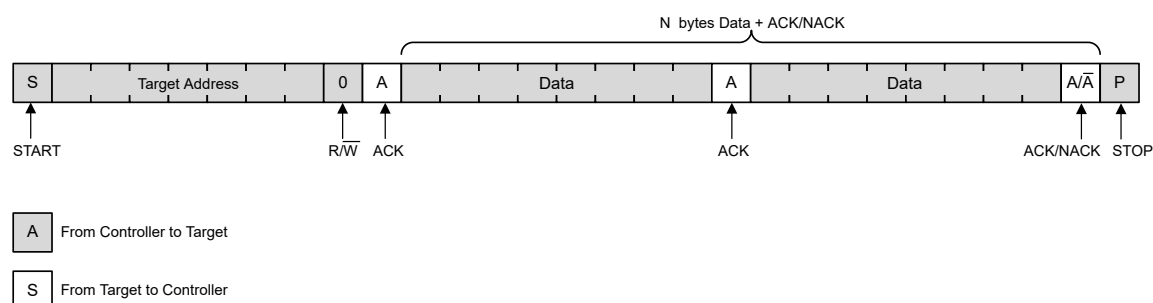


图 3-7 写操作帧格式

读操作

控制器以起始条件（S）和地址帧（R/W=1）发起一次读请求。地址被确认（ACK）后，控制器立即转变为接收器，目标设备转变为发送器。随后，目标设备控制 SDA 线串行输出数据字节，控制器在每个字节后的第 9 个时钟周期回复 ACK。在接收最后一个字节后，控制器必须回复 NACK，紧接着产生停止条件（P）以终止传输。

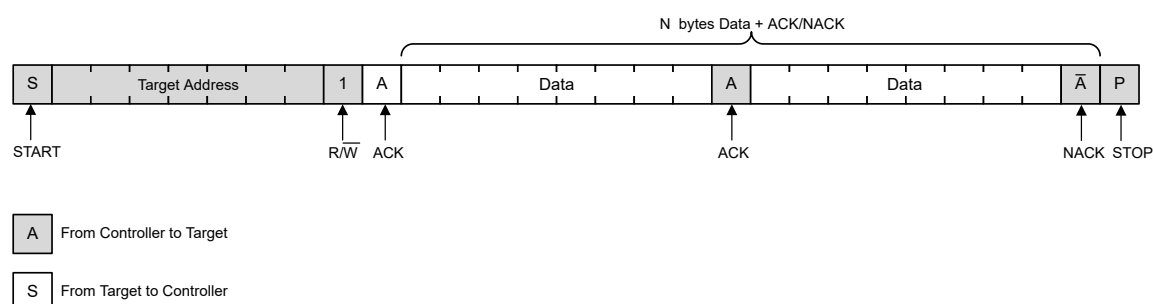


图 3-8 读操作帧格式

先写后读操作

若目标设备具有多个数据或配置寄存器，读取操作前可能需要先写入寄存器地址以指定读取目标。控制器首先以写模式（R/W=0）发起传输，发送目标地址及一个或多个数据字节，通常为待读寄存器指针，也有的描述为 Command Byte。随后，控制器不产生停止条件，而是发送一个重复起始条件，并再次发送同一目标设备的

地址，但将 R/W 取“1”以切换为读模式。之后流程与上述读操作完全相同。此模式通过重复起始条件维持总线控制权，实现了读写方向的无缝切换。

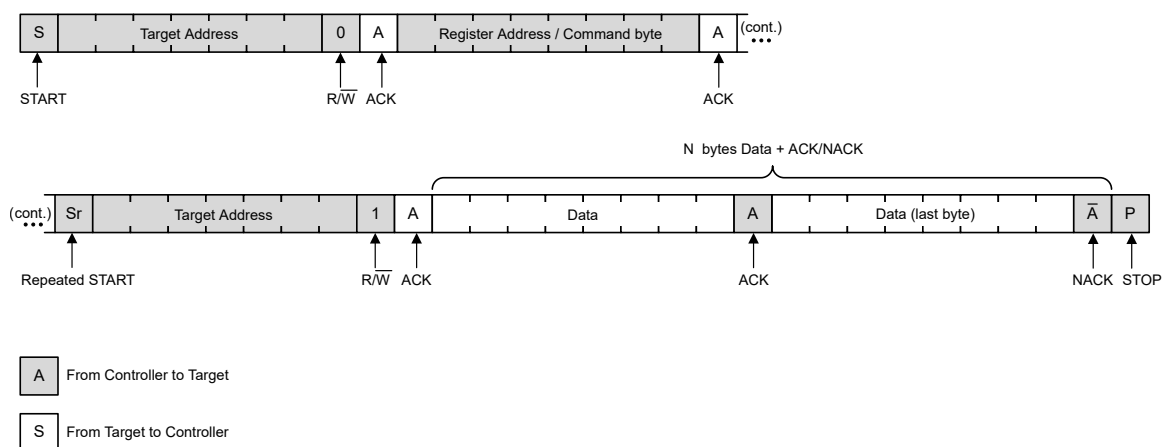


图 3-9 先写后读操作帧格式

3.2.2 连续读写与地址自动递增

在连续读写多个数据字节时，为提高传输效率，许多 I²C 器件支持地址自动递增功能。此功能指在一次 I²C 事务内，当控制器在不发送停止条件的情况下连续写入或读取多个数据字节时，目标器件内部的寄存器地址指针会在每个字节传输后自动递增，指向下一顺序寄存器地址。自动递增的核心优势在于提升总线利用率。它避免了为每个数据字节重复发送地址字节和起止条件，显著减少了协议开销字节总数，从而在总线初始化或批量数据传输时节省了时间和控制器资源。

自动递增是器件的可选功能，并非 I²C 协议标准所强制。其具体实现和行为因器件而异，分为多种类型：

（1）不支持寄存器地址自增，该类器件在连续读写时，仅重复写入或读取当前寄存器；（1）默认启用型，该类器件在检测到连续数据帧时即自动递增寄存器地址，此外部分器件仅在特定寄存器组内循环递增；（2）寄存器主动控制型，该类器件需通过配置特定寄存器位来显式启用此功能。

寄存器地址自动递增行为（如是否递增，递增方向、是否跨页、是否循环）完全由器件厂商定义。在实际应用中，需要查阅具体器件的数据手册，以确认其具体行为规则。

3.2.3 I²C 读写案例

以 TPT29548L（单寄存器型）和 TPT29508（多寄存器型）为例，分别展示这两类 I²C 器件的读写命令格式。通常其他 I²C 器件的读写操作也遵循类似的结构，具体以手册为准。

(1) TPT29548L 读写案例

TPT29548L 这类器件仅包含一个寄存器，其通过在目标地址之后立即发送寄存器数据的方式进行写入操作，而无需专门针对寄存器指针进行操作。同样，进行读操作时，直接发送读写位为‘1’的地址字节后，即可接收来自器件的数据。

a) TPT29548L 写命令

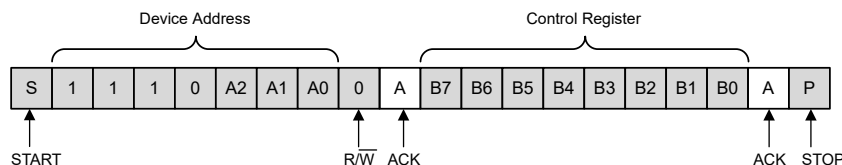


图 3-10 TPT29548L 写命令

b) TPT29548L 读命令

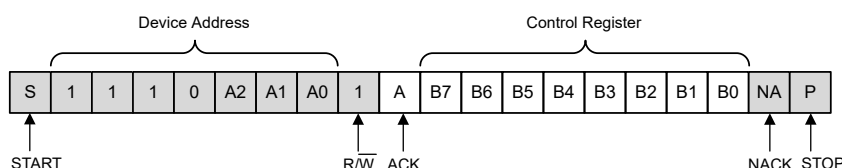


图 3-11 TPT29548L 读命令

(2) TPT29508 读写

TPT29508 内置多个寄存器，其写操作需在发送器件地址（写）后，先写入寄存器地址指针，再写入待配置的数据。读操作则需先通过一次写操作（器件地址写+寄存器地址指针）来设定待读取的寄存器地址，随后发起重复起始条件，再发送器件地址（读），之后即可从该器件连续读取寄存器数据。

a) TPT29508 写命令

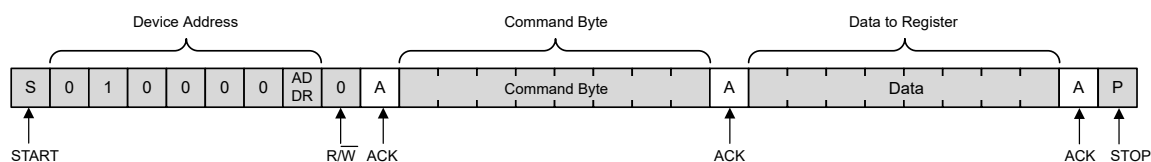


图 3-12 TPT29508 写命令

b) TPT29508 读命令

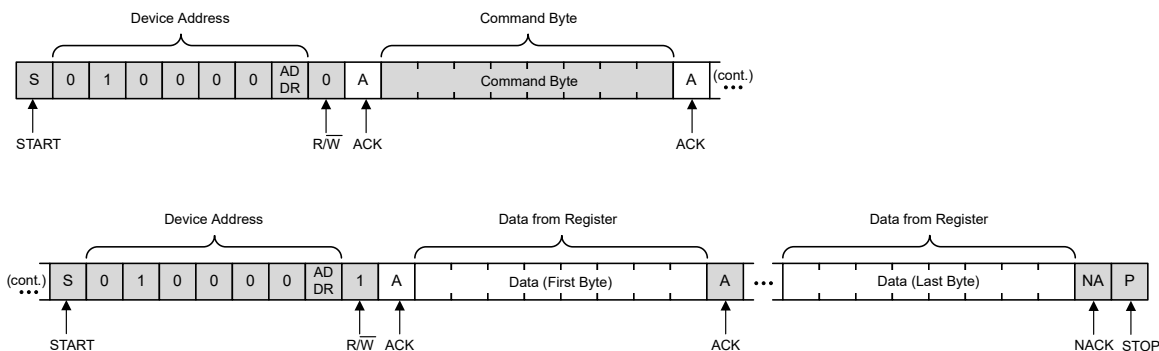


图 3-13 TPT29508 读命令 增加图例

3.3 I²C 地址

在 I²C 协议中，地址用于唯一标识总线上的目标设备（Slave/Target）。控制器在起始条件后发送的第一个字节即为地址字节，其中包含 7 位目标地址和 1 位方向位。标准的 I²C 地址为 7 位，理论上可寻址 128 个设备，但其中部分地址已被保留用于特殊用途，并非所有地址都可用于用户设备。这些保留地址格式主要为 0000 XXX 和 1111 XXX，用于广播呼叫、高速模式、器件 ID 访问等特定功能。需要注意的是，这些由保留地址调用的功能是 I²C 器件的可选选项，并非所有器件都支持。

模式	R/W	功能描述
0000 000	0	广播地址
0000 000	1	起始字节
0000 001	X	CBUS 地址
0000 010	X	保留用于不同总线格式
0000 011	X	保留供未来使用
0000 1XX	X	Hs-mode 模式 Code
1111 1XX	X	保留供未来使用
1111 0XX	X	10-bit 地址

3.3.1 I²C 器件地址类型

I²C 总线上每个目标设备都必须拥有一个唯一的地址以供控制器寻址。该地址通常由一个 7 位或 10 位的二进制数构成。在实际应用中，器件的地址配置方式多样，主要由硬件设计和制造商定义，可分为以下几种常见类型：

a) 固定地址

部分器件的 I²C 地址在出厂时已固化，用户无法更改。此类器件在总线上必须是唯一的，如果同一总线有多个相同地址器件需求，则可以通过 I²C 开关去分时访问，防止地址冲突。

b) 引脚可配置地址

这是最常见的 I²C 器件地址设置方式。该类器件的 7 位地址中，若干位由专用硬件地址引脚（例如 A0、A1、A2）的输入电平决定，这些引脚通常连接至 V_{CC} 或 GND。通过配置这些引脚的电平，可让同型号的多颗器件配置为不同地址，在同一 I²C 总线上共存。例如，TPT29555A 的地址格式为 ‘0100 A2 A1 A0’（其中 A2~A0 对应引脚电平的二进制值），因此最多可以实现 8 个器件挂载于同一条总线上。

此外，为了在地址引脚数量受限的情况下扩展更多地址，部分器件允许将地址引脚连接至 SCL 或 SDA 等 I²C 总线信号线，而不仅限于固定接 V_{CC} 或 GND。例如，TPA626 的两个地址引脚可配置为接 GND、接 V_{CC}、接 SCL 或接 SDA，从而提供 16 种不同的地址配置，显著增加同一总线上可挂载的器件数量。

还有一些器件，器件的地址由连接到特定引脚的外部上下拉电阻的阻值决定。例如，TPT29641 的地址引脚可以通过不同阻值电阻上拉或者下拉到 V_{CC} 和 GND，根据电阻阻值大小，映射到不同的地址。

c) 软件可配置地址

少数器件支持通过软件（即通过 I²C 命令写入内部寄存器）动态更改其自身在总线上响应的地址，从而在系统运行时重新分配地址空间。

SMBus 协议在 I²C 基础上定义了更完善的地址分配机制。SMBus 规范引入了地址解析协议（ARP），主控可通过标准 ARP 命令扫描总线、为未分配地址的设备动态指派唯一地址，并处理地址冲突。

I³C 协议将动态地址分配（DAA）作为协议核心特性。I³C 总线上所有目标设备上电后默认以静态地址或广播地址响应，主控通过 DAA 流程依次为每个设备分配唯一的 7 位动态地址。DAA 流程包含仲裁机制，可对多个同型号器件根据仲裁结果依次进行动态地址分配。

3.3.2 10-Bit 地址

为扩展寻址空间，I²C 协议还定义了 10 位地址模式。它利用保留地址 ‘11110XX’ 中的最后两位作为扩展地址的最高两位，与后续一个完整字节共同组成 10 位地址。进行 10 位地址通信时，控制器首先发送包含上述保留格式和写方位的地址字节，总线上所有高两位匹配的设备都会应答；接着发送低 8 位地址，此时唯一地址完全匹配的目标设备会继续应答，完成寻址。若需读取数据，控制器随后可发送一个重复起始条件，并再次发送相同的保留地址字节，但将方向位改为读，即可启动数据读取流程。此机制在有限的保留地址段内，将寻址能力从 128 个扩展到了 1024 个。

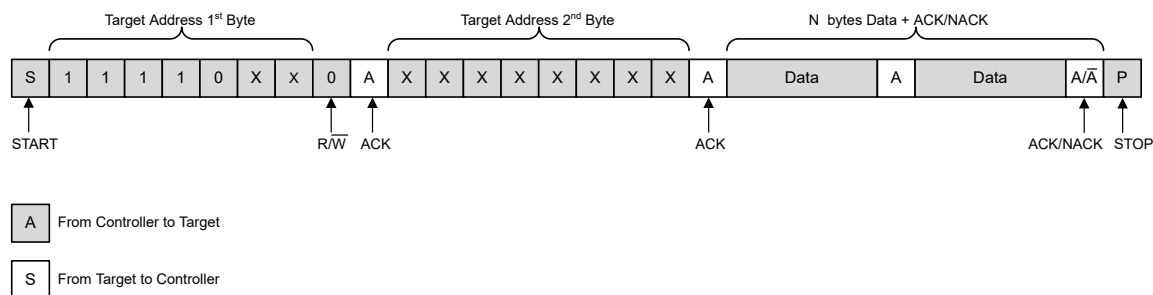


图 3-14 10-Bit 地址写操作

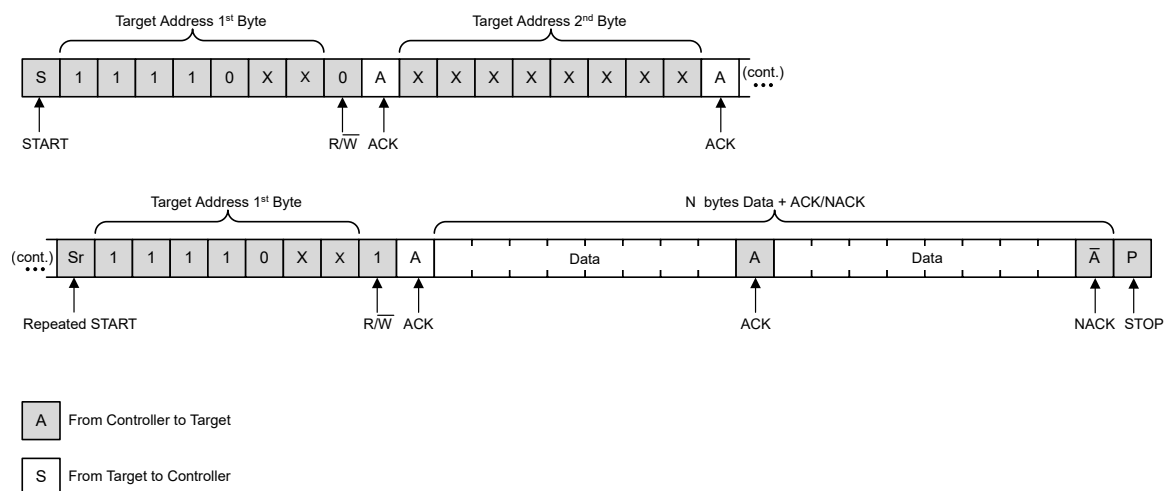


图 3-15 10-Bit 地址读操作

3.3.3 I²C 地址冲突

I²C 地址冲突是指在同一总线上挂载了两个或更多具有相同 I²C 地址的器件，导致控制器无法对其进行唯一寻址和正确通信。解决地址冲突、扩展总线挂载能力的一个有效方案是使用 I²C 多路复用器或开关器件，例如 TPT29548L。该器件本身作为一个具有唯一地址的从设备，其内部包含多个双向开关，可将上游的控制器总线切换连接到多个下游子总线上。控制器通过 I²C 命令来选择接通或断开特定下游总线，从而实现通道隔离和扩展地址空间。详细介绍见 [I²C 开关](#) 章节。

3.4 I²C 可选特性

I²C 协议除了起始条件、停止条件、应答等所有配置下均强制支持的基本特性外，还定义了时钟同步、仲裁、时钟延展、10 位地址、广播地址、软件复位和起始字节等其它特性，这些特性的强制性与适用性取决于系统配置要求。各特性的具体适用情况如下表所示。

表 3-1 I²C 总线强制与可选特性

特性	配置		
	单控制器（单主机）	多控制器（多主机）	目标设备（从机）
起始条件（START）	强制	强制	强制
结束条件（STOP）	强制	强制	强制
应答（ACK）	强制	强制	强制
时钟同步	不适用	强制	不适用
仲裁	不适用	强制	不适用
时钟延展	可选	可选	可选
7-bit 地址	强制	强制	强制
10-bit 地址	可选	可选	可选
广播地址	可选	可选	可选
软件复位	可选	可选	可选
START 字节	不适用	可选	不适用
设备 ID	不适用	不适用	可选

3.4.1 时钟同步

I²C 总线支持多主机架构，允许多个控制器共享同一总线。当两个或以上控制器在总线空闲状态下同时发起传输时，会产生总线竞争。此时，需通过时钟同步与仲裁机制来决定哪个控制器获得总线控制权，并确保数据传输不被破坏。在单控制器系统中，则不需要时钟同步和仲裁。

时钟同步通过 I²C 的 SCL “线与”实现。当多个控制器同时发起传输时，一旦 SCL 被拉低，所有控制器开始对各自的低电平周期进行计时。在此期间，即使某个控制器计时结束试图释放时钟，只要仍有其他控制器的低电平周期未结束，SCL 线就将一直被保持为低。当所有控制器的低电平周期都结束后，SCL 线才被释放为高，所有控制器随即开始对高电平周期计时。第一个完成高电平计时的控制器会再次将 SCL 线拉低。同步后 SCL 时钟的低电平周期由所有控制器中最长的低电平周期决定。同步后 SCL 时钟的高电平周期则由所有控制器中最短的高电平周期决定。通过此机制，多个控制器在竞争总线时能自动生成一个统一的、时钟周期被“拉伸”的同步 SCL 信号，为后续的逐位仲裁创造了时钟条件。

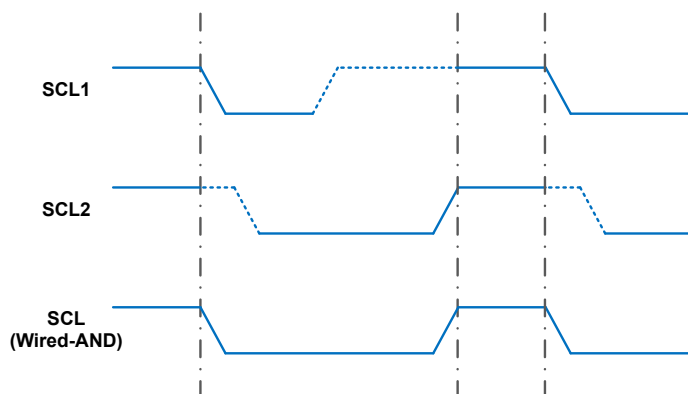


图 3-16 时钟同步

3.4.2 仲裁

与时钟同步类似，仲裁是仅在系统使用多个控制器时才需要的协议部分。它与时钟同步协同工作，确保在多个控制器同时发起传输时，具有非破坏性以及确定唯一控制权。仲裁过程仅控制器参与，目标设备不参与。控制器在总线空闲时启动传输，当两个或多个控制器同时产生有效的起始条件时，即触发仲裁。仲裁过程逐位进行，在每个数据位 SDA 进行线与，控制器在 SCL 为高电平期间检查 SDA 线的实际电平是否与自身试图发送电平一致。如果所有控制器发送的数据位完全相同，它们就可以无冲突地继续传输。一旦某个控制器尝试驱动高电平“1”但检测到 SDA 为低电平“0”，即表明其失去仲裁，该控制器必须立即关闭其 SDA 输出驱动器，停止驱动总线。赢得仲裁的控制器则继续完成整个事务。失去仲裁的控制器可以等待总线空闲后再重新尝试发起通讯。

图 3-17 展示了两个控制器的仲裁过程。

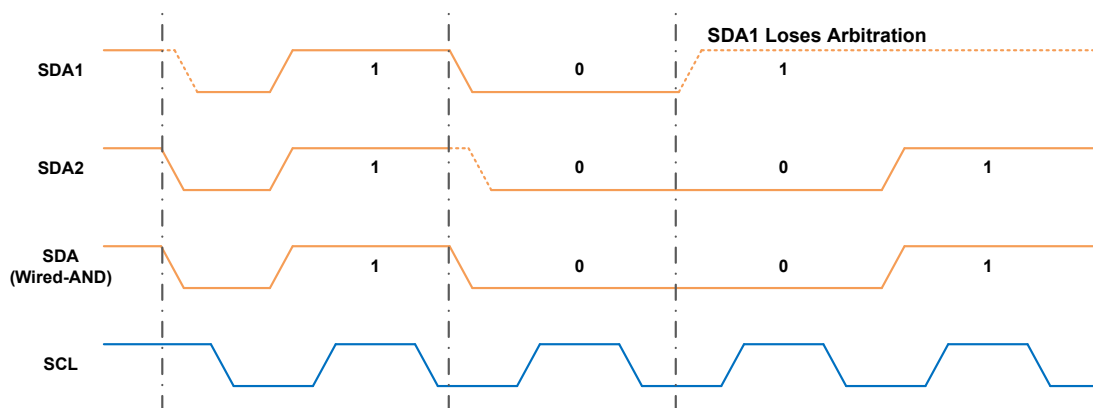


图 3-17 仲裁

3.4.3 时钟延展

时钟延展是由目标设备通过主动拉低并保持 SCL 线来暂停一个总线事务。由于 I²C 总线线与逻辑，SCL 将保持为低且无法被控制器释放。例如，目标在完成某些其他功能（例如处理内部中断）之前无法接收或传输另一个完整的字节数据，它可以将时钟线 SCL 保持在低电平，迫使控制器进入等待状态。然后，当目标准备好接收另一个字节数据并释放时钟线 SCL 时，数据传输将继续进行。

时钟延展可分为字节级延展和位级延展。

字节级延展是当目标设备接收或发送完一个字节后，若需要更多时间处理（如存储数据或准备下一字节），可在应答位之后将 SCL 拉低，迫使控制器等待，直至目标设备就绪并释放 SCL。

位级延展是在字节传输内部，微控制器等设备可通过延长每个 SCL 时钟的低电平周期来动态降低总线时钟频率，使其适配自身的实时处理能力。

时钟延展是可选的，需要主机和从机同时支持才可以实现。事实上，大多数目标设备不包含 SCL 驱动器，SCL 仅作为输入，因此它们无法延展时钟。

3.4.4 广播地址

广播地址是 I²C 协议中一个重要的保留地址，其 7 位地址为 ‘000 0000’（0x00），用于控制器向总线上的所有目标设备同时发起通信。当控制器发送此地址时，所有能识别该地址的设备都会进行监听。需要注意的是，并非所有设备都被强制要求响应此地址；若设备不支持或不需要处理广播数据，它可以忽略该地址而不发送应答信号。广播通信的事务格式通常为两字节序列。第一字节为广播地址（0x00），第二字节则用于定义具体的操作指令或地址信息，例如实现软件复位，编程从机地址等功能。

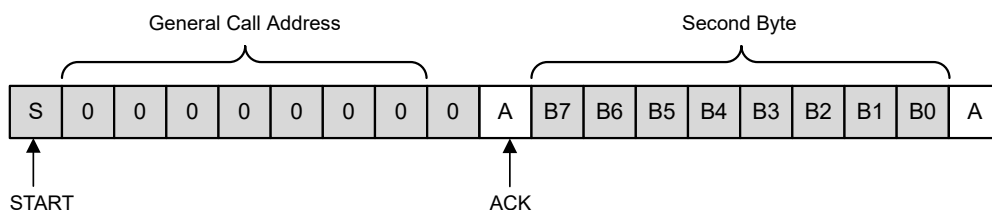


图 3-18 广播地址

3.4.5 软件复位

部分 I²C 器件支持软件复位功能，可以通过总线命令触发一个或多个目标设备执行复位操作。控制器发送一个两字节的特定命令序列，第一个字节为广播地址 0x00，第二个字节为命令代码 0x06。总线上所有支持并响应此广播命令的设备，在成功接收该序列后，将复位为上电默认状态。此功能也为可选，并非所有设备都会响应。

软件复位命令一般按照以下步骤和规则：

1. 控制器在总线上发起起始条件（START）。
2. 随后发送保留的通用广播地址 “000 0000” 和 $\overline{R/W}$ 位为 0（写操作），即完整地址字节为 0x00。
3. 支持该广播地址的器件将应答（ACK）。如果之前的地址字节 $\overline{R/W}$ 位设置为 1，器件将 NACK。
4. 在广播地址成功获得应答后，控制器发送一字节的数据 0x06。如果发送的数据字节非 0x06，器件将不会应答，后续也不会执行复位。如果发送的数据超过 1 字节，则不会再应答更多字节，且器件会忽略当前命令，将其视为无效。
5. 0x06 字节发送完成后，控制器发送一个停止条件来结束软件复位序列。器件会忽略重复启动条件，不执行复位。

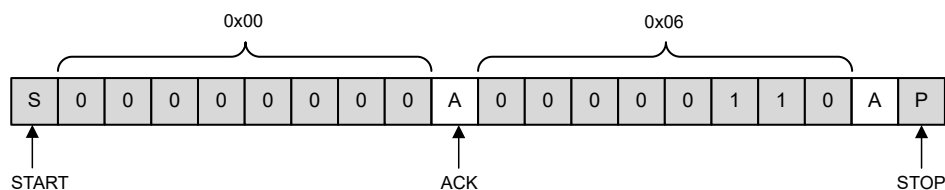


图 3-19 软件复位命令

3.4.6 START 字节

START 字节（0000 0001）是 I²C 协议中一个为特殊目的而保留的地址，它不是一个用于寻址具体设备的地址，而是一个辅助机制。作用是帮助那些没有硬件 I²C 接口、只能靠软件轮询总线的控制器，能可靠地检测到一次 I²C 传输的开始，从而进行后续的多控制器同步和仲裁。没有硬件 I²C 接口的控制器，通常会用较低的采样率去检测总线。这个低采样率很可能会错过短暂的 START 条件。因此其他控制器在发送 START 条件后，先不直接发送目标地址，而是先发送 Start Byte，所有从机都不应答（ACK）该字节。而当 SDA 线被拉的低连续 7 个 SCL 周期期间，低速率采样的控制器很容易捕捉到这个持续的低电平，从而切换到高速率采样模式。控制器之后发送一个重复 START 条件，随后高速控制器与低速控制器之前实现同步。START 字节仅在多控制器系统中，作为可选功能。

3.5 I²C 总线挂死和恢复

I²C 总线挂死是一种异常状态，指总线上的 SDA 或 SCL 线被意外锁定在低电平，导致无法通信。SDA 挂死最常见原因包括目标设备正在驱动总线为低时，控制器发生异常复位或者时钟丢失，此时目标设备将停在当前状态并等待新的时钟。SDA 挂死可能发生在两种位置：其一，目标设备在应答（ACK）位时，控制器在异常复位，此时目标设备将维持 ACK 状态，表现为 SDA 被持续拉低；其二，在读操作中，在目标设备应答并开始输出数据，当数据恰好为 0 时控制器异常复位，同样会导致 SDA 被目标设备锁定为低电平。

总线恢复的一种方法是继续生成时钟序列以完成被中断的传输。当检测到 SDA 线被持续拉低超过一定时间后，由控制器主动在 SCL 线上连续产生 0~9 个时钟脉冲。从而使仍处于“发送中”状态的目标设备完成当前字节，并释放 SDA 线。

- 如果总线挂死发生在向目标设备写入，目标设备 ACK 期间，控制器复位后需要 1 个 CLK 来使目标设备退出 ACK 状态，释放 SDA 总线，如图 3-20 所示。
- 如果总线挂死发生在从目标设备读取数据，目标设备正在输出‘0’时，如果当前为一字节数据的第 N 位，则控制器复位后最多（假如剩余位全为‘0’）需要 8-N 个 CLK 来使目标设备结束发送状态，释放 SDA 总线，如图 3-21 所示。
- 如果总线挂死发生在从目标设备读取数据，地址字节（R/W=1）后的 ACK 位置，则控制器复位后最多需要 9 个 CLK（最恶劣的情况下 1 个应答位+8 个均为‘0’的数据位后总线才能释放）来使目标设备结束发送状态，释放 SDA 总线。

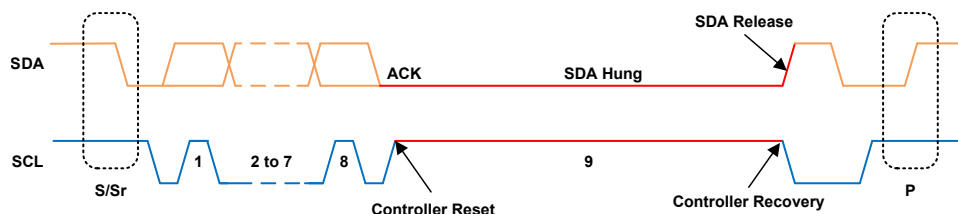


图 3-20 写操作时 ACK 期间挂死和恢复

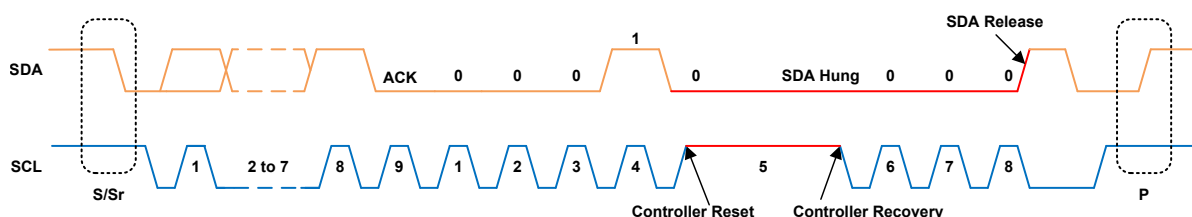


图 3-21 读操作时目标设备返回数据期间挂死和恢复

此外，还可以采用控制目标设备的复位引脚完成芯片复位。对于一些没有硬件复位的芯片，也可以使用重新上下电的方法，利用设备的上电复位电路强制清除其内部状态复位。

SCL 挂死相对少见，对于不支持时钟延展的目标设备，其 SCL 仅作为输入，无输出驱动能力。因此 SCL 挂死，原因一般是控制器本身发生异常。

4. I²C 速率模式

I²C 协议定义了多种总线速率模式，分别为：标准模式（Standard-mode, Sm）、快速模式（Fast-mode, Fm）、快速增强模式（Fast-mode Plus, Fm+）、高速模式（High-speed mode, Hs-mode）以及超快速模式（Ultra Fast-mode, UFm）。除超快速模式外，其余四种双向模式均向下兼容，而低速率模式的设备不能用于更高模式的总线，否则将因无法满足时序而导致不可预知的状态。不同模式的最大传输速率与方向如下表所示：

I ² C 模式	最大速率	数据方向
标准模式（Standard-mode, Sm）	100 kbit/s	双向
快速模式（Fast-mode, Fm）	400 kbit/s	双向
快速增强模式（Fast-mode Plus, Fm+）	1 Mbit/s	双向
高速模式（High-speed mode, Hs-mode）	3.4 Mbit/s	双向
超快速模式（Ultra Fast-mode, UFm）	5 Mbit/s	单向

a) 标准模式

标准模式作为 I²C 总线最基础的速率模式，最高速率为 100 kbit/s。

b) 快速模式

快速模式将速率提升至 400 kbit/s，同时要求设备在 SDA 和 SCL 输入端集成尖峰抑制滤波器与施密特触发器，提高抗干扰能力。标准模式和快速模式这两种模式获得了 I²C 器件最广泛的支持。

c) 快速增强模式

快速增强模式进一步将速率提高至 1 Mbit/s，但对总线驱动能力提出了更高要求，使其在同样负载下能满足更为严格的时序规格。以上三种模式具有相同的协议、帧格式与逻辑电平，差别仅在于时序规格与器件硬件实现的不同。支持以上三种模式的器件，可以向下兼容更低速率的模式。

d) 高速模式

高速模式将数据率提升至 3.4 Mbit/s，并保留向下兼容能力。高速模式器件的内部电路结构有所不同，SCL 除了对 GND 的下拉 MOS 外，还存在内部上拉电流源，如图 4-1 所示。进入高速模式前，控制器必须在标准模式或快速模式速率下先发送一个特殊的 8 位控制器代码“0000 1XXX”，该代码不允许任何设备应答。其后控制器将使能 SCL 线上的内部电流源上拉电路，将 SCL 上升时间大幅缩短，总线随即切换至高速模式。在高速模式传输期间，时钟同步与多控制器仲裁均不再执行（仲裁已在控制器代码阶段完成），从而消除了协议层面的速度瓶颈。SCL 时钟的高/低电平比被固定为 1:2，以更容易满足建立与保持时间的需求。

高速模式的进入和退出如图 4-2 所示。控制器在标准模式或快速模式速率下先发送控制代码“0000 1XXX”后进入高速模式。当通讯帧以停止条件（STOP）结束时，将退出高速模式回到标准模式或快速模式。若以重复起始条件（Sr）发起下一帧，则器件仍保持在高速模式下继续通信。

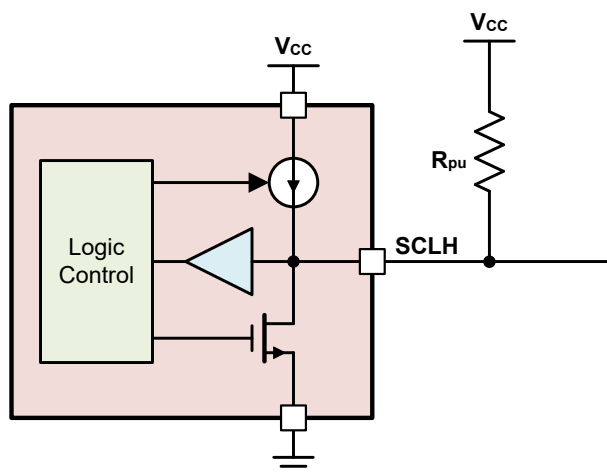


图 4-1 高速模式 SCL 物理层

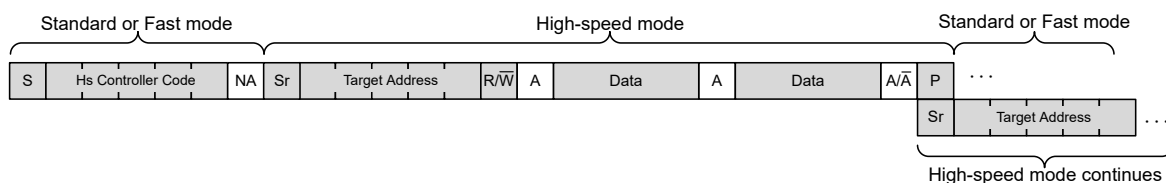


图 4-2 高速模式帧

e) 超快速模式

超快速模式采用推挽输出架构，最高传输速率可达 5 Mbit/s。该模式下，主控制器始终作为发送端，从器件仅具备接收能力。帧格式中移除了应答位（ACK），同时不再支持时钟延展、多主控制器仲裁等机制，仅保留起始条件、停止条件及数据字节的写操作。超快速模式器件无法向下与更低速率模式兼容，不能与双向 I²C 器件挂载于同一总线上。

5. I²C 电气和时序规范

5.1 I²C 电气规范

标准模式、快速模式和快速增强模式设备的 SDA 和 SCL 的 IO 引脚规范如下所示。

表 5-1 SDA 和 SCL 的 IO 引脚规范

符号	参数描述	条件	标准模式		快速模式		快速增强模式		单位
			最小值	最大值	最小值	最大值	最小值	最大值	
V _{IL}	低电平输入电压		-0.5	0.3V _{CC}	-0.5	0.3V _{CC}	-0.5	0.3V _{CC}	V
V _{IH}	高电平输入电压		0.7V _{CC}		0.7V _{CC}		0.7V _{CC}		V
V _{hys}	输入迟滞电压				0.05V _C		0.05V _C		V
V _{OL}	低电平输出电压	3mA 灌电流; V _{CC} > 2 V	0	0.4	0	0.4	0	0.4	V
		2mA 灌电流; V _{CC} ≤ 2 V	-	-	0	0.2V _{CC}	0	0.2V _{CC}	V
I _{OL}	低电平输出的流	V _{OL} = 0.4 V	3		3		20		mA
		V _{OL} = 0.6 V			6				mA
t _{of}	输出下降时间			250	20× (V _{CC} /5.5V)	250	20× (V _{CC} /5.5V)	120	ns
t _{SP}	输入滤波器抑制毛刺脉冲宽度 ⁽¹⁾				0	50	0	50	ns
I _i	输入电流	0.1V _{CC} < V _I < 0.9V _{CC}	-10	10	-10	10	-10	10	μA
C _i	引脚电容			10		10		10	pF

(1) 部分器件的 SDA、SCL 输入端均具备毛刺抑制功能，宽度小于 50ns 的毛刺可被输入滤波器有效滤除，防止其误触发总线逻辑。支持快速模式器件的可以向下兼容标准模式，即使运行在标准模式速率，输入滤波器依旧生效。

5.2 I²C 时序规范

I²C 总线标准模式、快速模式及快速增强模式的时序参数如下表所示。

表 5-2 I²C 时序规范

符号		参数描述	标准模式		快速模式		快速增强模式		单位
			最小值	最大值	最小值	最大值	最小值	最大值	
f _{SCL}	f _{scl}	SCL 时钟频率	0	100	0	400	0	1000	kHz
t _{HD;STA}	t _{sth}	(重复) 起始条件保持时间	4		0.6		0.26		μs
t _{LOW}	t _{scl}	SCL 时钟低电平周期	4.7		1.3		0.5		μs
t _{HIGH}	t _{sch}	SCL 时钟高电平周期	4		0.6		0.26		μs
t _{SU;STA}	t _{sts}	重复起始条件建立时间	4.7		0.6		0.26		μs
t _{HD;DAT}	t _{sdh}	SDA 数据保持时间	0		0		0		μs
t _{SU;DAT}	t _{sds}	SDA 数据建立时间	250		100		50		ns
t _r		SDA 和 SCL 信号上升时间		1000	20	300		120	ns
t _f		SDA 和 SCL 信号下降时间		300	20× (V _{CC} /5.5V)	300	20× (V _{CC} /5.0V)	120	ns
t _{SU;STO}	t _{sps}	STOP 条件建立时间	4		0.6		0.26		μs
t _{BUF}	t _{buf}	STOP 和 START 之间的总线空闲时间	4.7		1.3		0.5		μs
C _b		总线电容负载		400		400		550	pF
t _{VD;DAT}	t _{vd(data)}	数据输出建立时间		3.45		0.9		0.45	μs
t _{VD;ACK}	t _{vd(ack)}	ACK 输出建立时间		3.45		0.9		0.45	μs
V _{nL}		低电平噪声容限	0.1V _{CC}		0.1V _{CC}		0.1V _{CC}		V
V _{nH}		高电平噪声容限	0.2V _{CC}		0.2V _{CC}		0.2V _{CC}		V

I²C 总时序参数定义和说明如下。

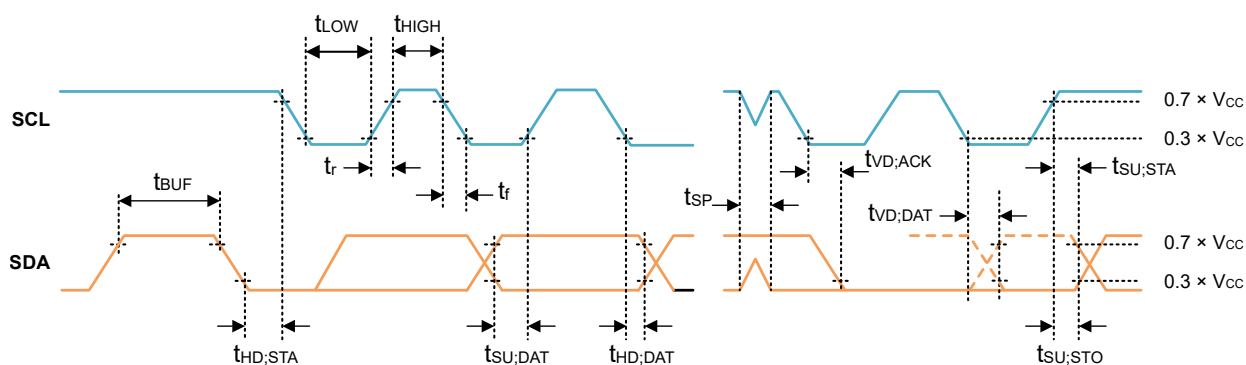


图 5-1 I²C 时序定义

a) fSCL

串行时钟线上允许的时钟频率，标准模式最高为 100kHz，快速模式最高为 400kHz，快速增强模式最高为 1MHz。

b) tHD;STA

从 SDA 下降沿（起始条件）出现后到 SCL 第一个下降沿之间的最小保持时间。该时间结束后，总线方可产生第一个时钟脉冲，以确保起始条件被可靠识别。

c) tLOW

信号处于低电平的最短持续时间。它与 t_{HIGH} 共同决定总线时钟频率上限。设备可通过拉长 t_{LOW} 实现时钟延展，以降低实际传输速率。

d) tHIGH

信号处于高电平的最短持续时间。

e) tSU;STA

在 SCL 上升沿之后，SDA 必须维持在高电平的最小稳定时间，方可被拉低以产生重复起始条件。该参数确保重复起始条件不会过早出现而被误判为数据位。

f) tHD;DAT

SCL 下降沿出现后，SDA 线上数据必须保持稳定的最短时间。标准模式、快速模式和快速模式增强版均允许 t_{HD;DAT} 最小为 0 ns。应用中要求 SCL 先降至 0.3V_{CC} 以下，SDA 才能进入 0.3V_{CC} 至 0.7V_{CC}。一般情况下，建议给控制器 t_{HD;DAT} 设置比 0ns 更高的裕量，例如 100ns，以满足不同器件或场景的需求，提高通讯的稳定性。

g) tSU;DAT

SCL 上升沿之前，SDA 线上数据必须稳定建立的最短时间。标准模式为 250ns，快速模式为 100ns，快速增强模式为 50ns。

h) t_r 和 t_f

t_r 是 SDA 和 SCL 信号从 $V_{IL(max)}$ 上升到 $V_{IH(min)}$ 所需时间, t_f 则是从 $V_{IH(min)}$ 下降到 $V_{IL(max)}$ 的时间。标准模式下 t_r 上限为 1000ns, t_f 上限为 300 ns。快速模式的 t_r 限定在 20~300 ns 之间, t_f 为 $(20 \times (V_{CC}/5.5V))$ ns 至 300 ns。快速模式增强版将 t_r 和 t_f 均限制在 120 ns 以内, 且输出级的下降时间 t_{of} 与总线下降时间 t_f 要求一致。若在引脚间串联电阻, 设计时需将其对 t_f 的贡献一并计入。

实际应用中如果主机驱动能力较强, 信号下降沿时间可能低于 20ns, 一般 I²C 目标器件也可以接受, 具体以手册描述或厂家答复为准。

i) $t_{SU};STO$

SLC 上升沿出现后, SDA 从低电平跳变为高电平之前所需的最小建立时间。该参数确保停止条件在 SCL 处于高电平期间被可靠识别, 防止 SDA 过早拉高导致无法识别到 STOP 条件。

j) t_{BUF}

定义为一次传输的停止条件与下次传输的起始条件之间 SDA 和 SCL 均处于高电平的最小持续时间。该参数保证了总线恢复至空闲状态并允许新的控制器接管总线。

k) $t_{VD};DAT$

读数据期间, 从 SCL 下降沿到 SDA 输出有效数据之间的最大允许时间。标准模式上限为 3.45 μs , 快速模式 0.9 μs , 快速增强模式为 0.45 μs 。

l) $t_{VD};ACK$

从 SCL 下降沿到 SDA 上应答信号有效输出的最大允许时间。

m) C_b

每条总线线路上允许的最大对地电容。标准模式和快速模式为 400pF, 快速增强模式为 550pF。

5.3 高速模式和超快速模式规范

高速模式与超快速模式是 I²C 协议中面向更高数据速率需求定义的两种速率等级。这两类器件在实际应用中相对少见, 其电气特性和时序数值参考协议规范文档。

6. I²C 开关

6.1 概述

I²C 开关（I²C Switch）是一类用于扩展 I²C 总线的器件，其片内从机寄存器可以控制上游总线与多路下游总线之间连接通路的开启或关断，实现一对多的总线拓扑管理。同时，I²C 开关具备电平转换功能，其内部 PassFET 架构允许上下游通道分别通过电阻上拉到不同总线电平，通道开启时实现自动双向电平转换，无需额外电平转换器件。

I²C 开关可适用于以下场景：

1. 当系统中存在多个地址完全相同的目标器件时，将各器件置于 I²C 开关不同下游通道，控制器分时选通访问，从根本上解决地址冲突。
2. 在复杂系统中总线容性负载较重时，可通过 I²C 开关将总线划分为若干独立区段，各段负载独立控制，减小单段等效电容，使总线更易满足目标速率模式的时序要求。
3. I²C 开关可将总线按功能或物理位置进行分段隔离管理。当某通道上的器件发生故障并将总线挂死时，可通过硬件复位切断该通道，实现故障隔离。
4. I²C 开关可支持混合速率与多电压域总线共存。不同通道可挂载不同速率模式的器件，控制器切换通道以应对不同速率通信。不同通道可分别上拉至不同总线电压，实现跨电压域通信。

6.2 3PEAK I²C 开关系列

3PEAK 提供覆盖 2 通道、4 通道及 8 通道的 I²C 开关产品系列。部分型号集成中断输出（ $\overline{\text{INT}}$ ）功能， $\overline{\text{INT}}$ 引脚为开漏输出，多个中断源线与后传播给上游控制器。工作电压覆盖 1.8V 至 5.5V，适配主流嵌入式系统供电范围，I/O 耐压支持跨电压域应用。

特性	TPT29548L	TPT29546L	TPT29548	TPT29546	TPT29545	TPT29641
通道数量	1:8	1:4	1:8	1:4	1:4	2:1
最小工作电压	1.65V	1.65V	2.3V	2.3V	2.3V	2.3V
最大工作电压	5.5V	5.5V	5.5V	5.5V	5.5V	3.6V
电平转换	✓	✓	✓	✓	✓	✓
硬件复位	✓	✓	✓	✓	✓	✓
中断功能					✓	✓
地址引脚数量	3bit	3bit	3bit	3bit	2bit	4bit

6.3 I²C 开关工作原理

I²C 为一般具有一路 SDA/SCL 上游通道和多路 SDA_n/SCL_n 下游通道，且内部包含一个片内寄存器，通过给寄存器发送 I²C 命令，来控制下游通道的开启与关断。片内寄存器直接通过上游 SDA/SCL 总线进行读写，因此无需额外的控制引脚。I²C 开关的典型结构框图如 1 所示。

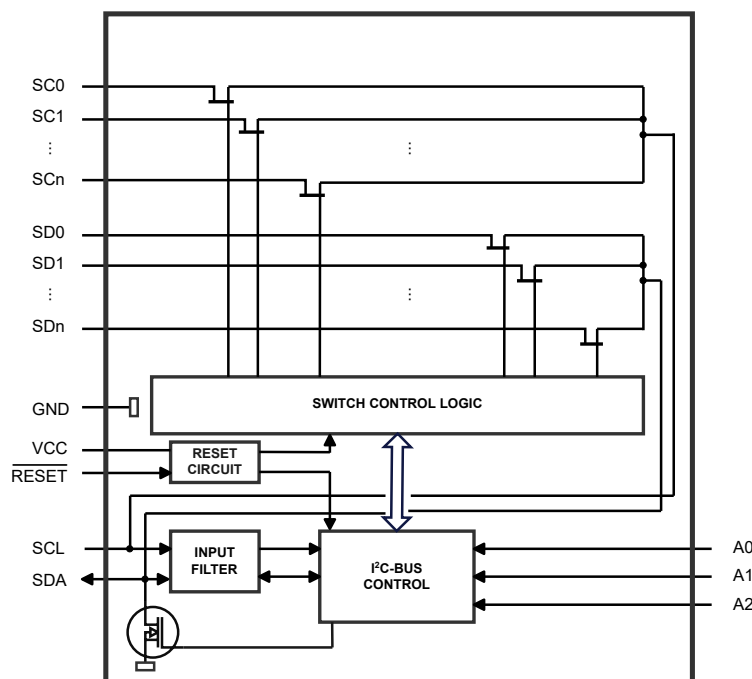


图 6-1 I²C 开关结构框图

I²C 开关的每个通道由衬底接地的 NMOS 构成，也被称为 PassFET，如图 6-2 所示。MOS 的栅极电压由寄存器进行控制。

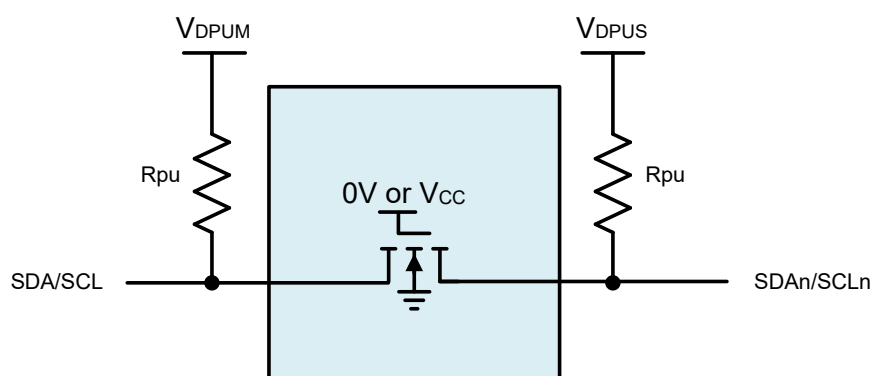


图 6-2 I²C 开关 PassFET 架构

a) 通道选通

I²C 开关通过 I²C 命令控制各通道的通断。器件内部仅包含一个控制寄存器，无需指定寄存器地址，通信协议简洁高效。以 TPT29548L 为例，控制寄存器如图 6-3 所示。寄存器的每一位映射到一个下游通道：写入 1 使能对应通道，写入 0 断开对应通道。控制位置 1 时，对应通道的 MOS 栅极被驱动至 V_{CC}，此时通道开启，且兼

具双向传输与电平转换能力。控制位清 0 时，MOS 栅极被驱动至 0V，MOS 在整个输入电压范围内完全关断，上下游通道彻底隔离。器件上电或复位后，控制寄存器默认全零，所有通道处于断开状态。

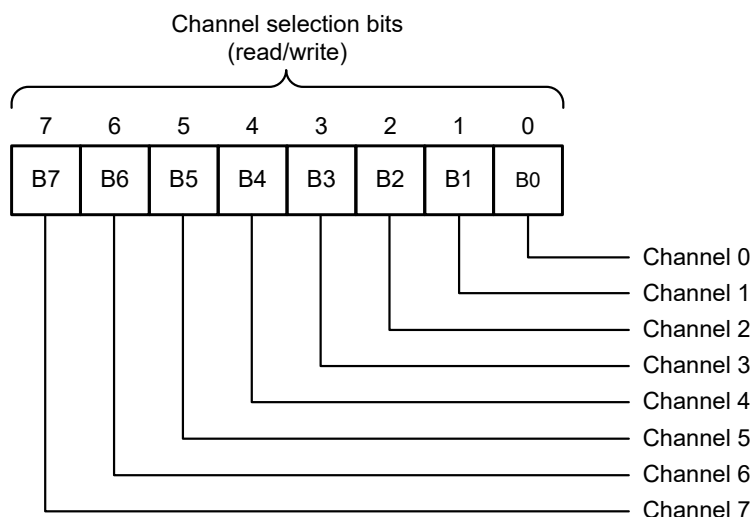
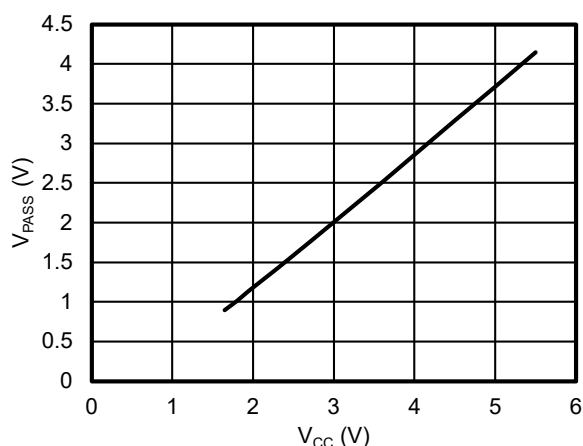


图 6-3 TPT29548L 控制寄存器

b) 电平转换机制

当通道被使能时，内部 MOS 栅极被驱动至 V_{CC} 。此时 MOS 的导通与否取决于其输入电压：当输入电压低于 V_{CC} 减去 MOS 开启阈值电压 V_{TH} ，即 V_{PASS} ($V_{PASS} = V_{CC} - V_{TH}$) 时，MOS 导通。

因此，当 I²C 输入低电平信号，输入电压接近 0V，远低于 V_{PASS} ，MOS 保持导通，低电平可经低阻路径传至另一侧。当输入为高电平时，电平超过 V_{PASS} ，MOS 关断呈高阻态，两侧总线互相隔离，各自通过本侧上拉电阻拉到对应的高电平电压。正是这一机制，使 PassFET 无需方向控制即可实现双向电平转换。



不同 I²C 开关器件的 MOS 导通阈值 V_{TH} 不同，因此 V_{PASS} 曲线也不一致，实际应用参考芯片数据手册。或者可以参考手册中 $V_{O(sw)}$ 参数，其数值与 V_{PASS} 相同。

由此可见，实现正常电平转换的前提是：上游上拉电压 V_{DPUM} 与下游上拉电压 V_{DPUS} 均须大于 V_{PASS} 。若任一电压低于 V_{PASS} ，MOS 在高电平期间无法关断，导致电平转换失效。

以 TPT29548L 为例，如果上游总线电压 $V_{DPUM} = 3.3\text{ V}$ ，下游总线电压 $V_{DPUS} = 5\text{ V}$ 。如果芯片 V_{CC} 供电选择 5 V ，则对应的 V_{PASS} 电压为 3.6 V 。此时 V_{DPUM} 低于 V_{PASS} ，高电平期间 MOS 无法关断，最终低压侧总线降被高压侧拉高，直到 MOS 关闭。开关不仅丧失电平转换功能，实际总线电压还可能超出低压侧器件的耐压限值，造成潜在损害。正确的做法是 V_{CC} 选取 4.63 V 以下的供电，即对应的 V_{PASS} 低于 3.3 V 。更便捷的做法，可以直接使用低压侧 3.3 V 电源作为 V_{CC} 供电，由于 V_{PASS} 低于 V_{CC} ，因此一定能满足 V_{DPUM} 和 V_{DPUS} 均低于 V_{PASS} 的需求。

6.4 器件寻址与 I²C 读写

a) 器件地址

I²C 开关拥有唯一的 7 位从机地址。以 TPT29548L 为例，其地址高 4 位固定为“1110”，低 3 位由硬件地址引脚 A2、A1、A0 的电平决定。用户通过外部上拉或下拉电阻将地址引脚接至 V_{CC} 或 GND，可获得 8 个可选地址，因此最多允许同一 I²C 总线上最多挂载 8 片同型号开关器件而互不冲突。地址引脚内部未集成上下拉电阻，设计时必须为其提供确定的外部电平，不得悬空。

b) 控制寄存器

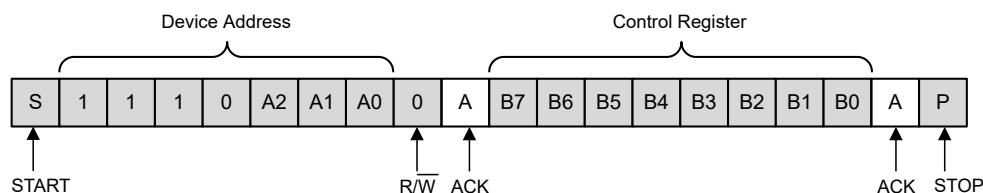
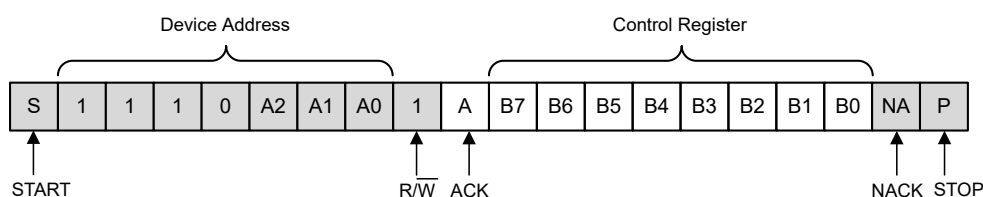
I²C 开关器件仅包含一个寄存器，其通过在目标地址之后立即发送寄存器数据的方式进行写入操作。控制器在发送器件地址并收到 ACK 后，紧接着写入一个 8 位控制字节。寄存器的每一位对应一个下游通道：置 1 使能，清 0 断开。上电或硬件复位后，控制寄存器默认值为“0x00”，所有通道处于断开状态。通道切换并非在控制字节的 ACK 之后立即生效，而是在控制器发出停止条件之后才正式使能。这一延迟机制确保通道切换瞬间所有 SCx/SDx 线路均已处于高电平状态，避免连接时产生虚假的起始条件或数据脉冲，保证下游器件状态机不受干扰。若控制器在一次传输中写入多个字节，后面的字节会覆盖前面的写入结果，因此仅最后一个字节生效。

控制寄存器可以支持同时使能多个通道。例如，TPT29548L 控制寄存器写入 B7=0、B6=1、B5=0、B4=0、B3=1、B2=1、B1=0、B0=0 时，通道 6、通道 3 和通道 2 同时选通，其余通道断开。多通道同时选通时，各下游通道在电气上彼此并联，控制器与所有选中通道上的器件共享同一总线，需要注意同时开启的各通道累加的总电容不能超过目标速率模式的 C_b 上限。

c) 寄存器读写

写控制寄存器：控制器发起起始条件（S），发送器件地址（ $\overline{R/W}$ 位为 0），收到 ACK 后直接发送控制字节，收到 ACK 后发送停止条件。器件仅有一个控制寄存器，无需发送寄存器地址。

读控制寄存器：控制器发起起始条件，发送器件地址（ $\overline{R/W}$ 位为 1），收到 ACK 后器件将当前控制寄存器的内容输出至 SDA 线。控制器读取一个字节后以 NACK 应答，随后发送停止条件结束读事务。读操作可随时查询当前通道选通状态，用于诊断或状态确认。


图 6-4 I²C 写命令

图 6-5 I²C 读命令

6.5 2:1 I²C 开关 (仲裁器)

除了 TPT2954x 这类 1: N 的 I²C 开关外，还有一类 2:1 的 I²C 开关，例如 TPT29641。其用于多路 I²C 上游控制器与单路下游总线通讯。其内部也为 MOS 架构，支持通道间电平转换。两路主机均可通过寄存器控制下游通道开启，TPT29641 还支持硬件仲裁功能，可以实现两路主机访问下游总线而不发生冲突导致数据丢失。先发出连接请求的主机获得下游总线控制权，发送放弃总线命令或者总线保留时间到期后断开连接；

TPT29641 工作流程如下：先发出连接请求的主机获得下游总线控制权，发送放弃总线命令或者总线保留时间到期后自动断开连接。后发出连接请求的主机可以被 TPT29641 保留请求并排队等待，直到之前的主机结束控制后，自动连接下游总线并触发中断通知主机。

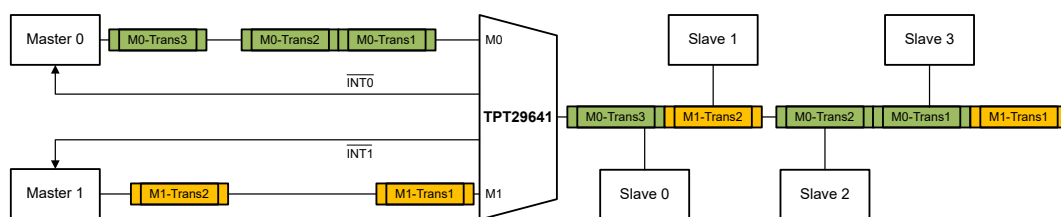


图 6-6 TPT29641 双主机仲裁通讯过程

此外，TPT29641 还支持总线挂死检测，总线初始化，总线复位，中断检测，双主机通讯等功能。

6.6 I²C 开关与模拟开关对比

IC 开关与模拟开关均可控制切换在多个信号通路，但二者在控制方式、电气特性和功能模式上存在本质差异。

I²C 开关内部集成片内从机控制逻辑，控制器可通过 I²C 总线命令控制各通道的开启与关断，无需占用额外 GPIO 资源。模拟开关则依赖外部使能引脚来控制通道选通，每增加一个通道选择位即需占用一个 GPIO，随着通道数增加，主控引脚开销与 PCB 走线复杂度线性增长。

I²C 开关基于 PassFET 架构，内部传输门为无源双向导通，两侧总线可分别通过各自的上拉电阻拉到不同总线电平，实现双向电平转换。模拟开关导通时两侧电压须处于同一电平范围，跨电压域使用时需增加电平转换电路。

I²C 开关通常内置多种功能：支持硬件地址引脚，允许多片开关共享同一 I²C 总线；部分器件具备复位输入引脚，可通过硬件信号将所有通道强制断开，作为总线恢复的手段；一些器件提供中断汇总输出，将下游多通道的中断信号合并为单一开漏输出线，减少控制器中断引脚占用。普通模拟开关仅为信号通路选择器，不包含上述功能。

对于涉及多电压域总线扩展、降低系统复杂度或节省主控 IO 资源等需求的 I²C 系统，I²C 开关相比模拟开关具有显著的系统集成优势。

6.7 应用注意事项

上拉电阻注意事项：

当使用 I²C 开关时，上游与下游每个通道均需配置上拉电阻。高电平期间 I²C 开关内部 MOS 关断，高电平需要通过电阻上拉实现。

低电平期间 MOS 为低 R_{on} 直接导通，因此控制器或目标设备输出低电平时，所有上拉电阻电流都会流向其内部下拉 MOS，可以等效为所有电阻并联，如图 6-7 所示。当总电流较大时，可能造成低电平抬升。因此建议每段总线根据各自的寄生电容选取合适的上拉电阻，如 I²C 上拉电阻选取节所述。访问下游通道时尽量只开启需要访问的通道，不要同时开启多个通道防止电容累加以及电阻并联导致下拉电流增大。

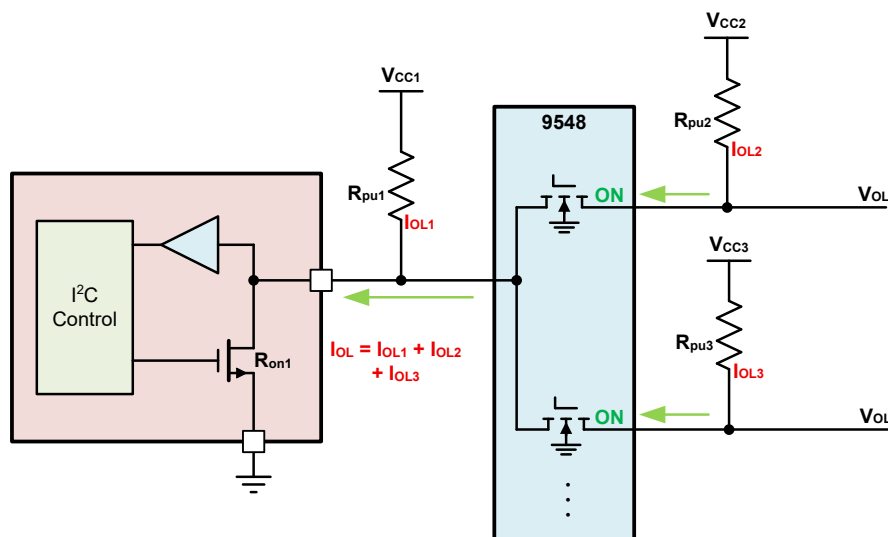


图 6-7 I²C 开关上拉电阻总电流

如果某个正在使用的通道没有加上拉电阻，其通道开启后输出通过 MOS 充电，引脚电压上升到 V_{PASS} 后 MOS 关闭，最终高电平只能上升到 V_{PASS} 无法达到高电平电压，可能导致无法正常通讯。

I²C 开关未使用的通道，建议通过 $10k\Omega \sim 100k\Omega$ 电阻上拉到 V_{CC} 或保持悬空。

7. I/O 拓展

7.1 概述

在嵌入式系统中，如果主控 GPIO 数量难以满足 LED 指示、按键扫描、电源使能、中断输入及拨码开关读取等板级外设的管理需求，可以使用 I/O 扩展芯片为系统提供额外的通用输入输出端口。部分 I/O 扩展芯片为 I²C 接口，控制器仅需 SDA 和 SCL 两根信号线即可读写远端 I/O 端口状态，可以有效拓展主控 IO 资源并降低系统复杂度。I²C I/O 扩展芯片与 I²C 系统高度耦合，常与 I²C 开关、缓冲器等协同构建板级互联方案。

7.2 3PEAK 选型

3PEAK 提供覆盖 4 位至 16 位的 I²C 扩展 I/O 芯片，适配不同规模和功能需求的系统。

特性	TPT29555A	TPT29535A	TPT29554A	TPT29536A	TPT29539AQ	TPT29508	TPT29516
I/O 端口数量	16	16	8	4	16	8	16
最小工作电压	1.65 V	1.65 V	1.65 V	1.65 V	1.65 V	1.65 V	1.65 V
最大工作电压	5.5 V	5.5 V	5.5 V	5.5 V	5.5 V	5.5 V	5.5 V
电平转换						✓	✓
硬件复位					✓	✓	✓
中断	✓	✓	✓		✓	✓	✓
内部上拉电阻	100kΩ	NULL	100kΩ	100kΩ	NULL	NULL	NULL
地址引脚数量	3bit	3bit	3bit	fixed	2bit	1bit	1bit

7.3 特性介绍

a) I/O 端口

所有端口均可配置为输入端口或输出端口。当端口配置为输入时，内部输出驱动级呈高阻态，引脚电平完全由外部电路以及芯片内部上/下拉电阻决定。当端口配置为输出时，内部推挽驱动级根据输出寄存器状态将引脚拉至 V_{CC} 或 GND，此时引脚与电源轨之间存在低阻抗通路，具备较强的驱动能力。

TPT295xx 系列芯片的 I/O 端口，可以支持 5.5V 耐压，并具备防倒灌结构。即当 V_{CC} 掉电或者后上电，而输入端口被外部信号拉高时，也不会产生从 IO 端口到 V_{CC} 的电流倒灌。

b) 中断

端口配置为输入模式时，其电平翻转会触发中断，芯片的 $\overline{\text{INT}}$ 引脚将拉低，向控制器发出通知。中断清除通过读取输入端口寄存器完成，读操作完成后 $\overline{\text{INT}}$ 释放回高电平。或者当输入电平恢复之前值，中断也会自动清除。若读取期间又有新的电平变化发生，该事件被保留， $\overline{\text{INT}}$ 保持低电平直至下一次读取将其清除。

$\overline{\text{INT}}$ 为开漏输出，需外接上拉电阻至 V_{CC} ，多个器件的中断引脚可线与共用一根中断线。配置为输出的端口不参与中断产生。将端口由输出切换为输入时，若引脚当前电平与输入端口寄存器内容不一致，可能产生伪中断，软件初始化阶段应加以处理。

另外在芯片上电期间，如果输入电平不确定，或者输入电平上电存在延时（例如端口外接较重容性负载），可能导致芯片检测到输入端口电平变化从而触发中断，导致上电完成后 $\overline{\text{INT}}$ 为低。这种情况下，可以在上电完成后或者进行系统配置前，主动读取 I/O 拓展芯片的输入寄存器，从而复位中断。

c) 复位

复位引脚在拉低可以触发器件复位，芯片寄存器和状态机恢复默认值。若系统中未使用复位功能，该引脚须通过上拉电阻接至 V_{CC} ，不得悬空。不含复位引脚的型号则需依赖掉电复位实现状态恢复。此外，部分 I/O 拓展芯片还支持软件复位功能。

d) 电平转换

TPT29508 和 TPT29516 采用双电源域设计，I²C 总线侧 V_{CCI} 与 I/O 端口侧 V_{CCP} 独立供电。输入/输出端口逻辑高电平由 V_{CCP} 决定，与 I²C 总线电压解耦。对应主控电平与外部 IO 电平不一致的应用场景，可以省去额外的电平转换器件。

例如，I²C 总线侧主控工作于 1.8 V，I/O 端口侧外设电平为 3.3 V，可使用该类双电源 I/O 拓展芯片。当未使用此功能时， V_{CCI} 与 V_{CCP} 可接同一电源。

e) 地址引脚

I/O 拓展芯片可通过硬件地址引脚设定 I²C 从机地址。地址引脚须外接确定电平，不得悬空。不同型号提供不同数量的地址引脚。

例如，TPT29555A 地址如下所示：

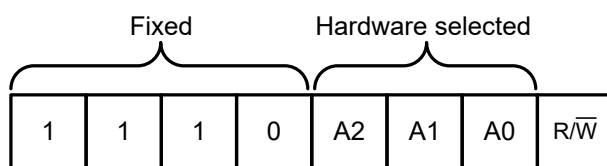


图 7-1 TPT29555A 地址

7.4 寄存器描述与 I²C 读写

I/O 扩展芯片内部包含四类寄存器，分别为：输入寄存器、输出寄存器、极性反转寄存器和输入输出模式配置寄存器。分别用于读取输入电平、设定输出电平、反转输入极性和配置端口方向。

对于 16 位 I/O 扩展芯片，每组寄存器包含两个 8 位寄存器，分别对应 Port0（P0_0~P0_7）和 Port1（P1_0~P1_7），共计 8 个寄存器。对于 8 位 I/O 扩展芯片，每组仅含一个 8 位寄存器，共计 4 个寄存器。

例如，TPT29555A 为 16 位 I/O 扩展芯片，其寄存器映射如下。

表 7-1 TPT29555A 寄存器

寄存器地址（命令字节）	寄存器描述	特性	默认值
0x00	Port0 输入寄存器	只读	0xXX
0x01	Port1 输入寄存器	只读	0xXX
0x02	Port0 输出寄存器	读/写	0xFF
0x03	Port1 输出寄存器	读/写	0xFF
0x04	Port0 极性反转寄存器	读/写	0x00
0x05	Port1 极性反转寄存器	读/写	0x00
0x06	Port0 输入输出模式配置寄存器	读/写	0xFF
0x07	Port1 输入输出模式配置寄存器	读/写	0xFF

例如，TPT29554A 为 8 位 I/O 扩展芯片，其寄存器映射如下。

表 7-2 TPT29554A 寄存器

寄存器地址（命令字节）	寄存器描述	特性	默认值
0x00	输入寄存器	只读	0xXX
0x01	输出寄存器	读/写	0xFF
0x02	极性反转寄存器	读/写	0x00
0x03	输入输出模式配置寄存器	读/写	0xFF

这四类寄存器的定义和功能如下：

输入寄存器：只读寄存器，反映对应端口的实际引脚电平。无论端口配置为输入还是输出，该寄存器均返回引脚当前逻辑值。写操作对该寄存器无效，上电默认值由外部引脚电平决定。

输出寄存器：控制配置为输出的端口的驱动电平，写 1 输出高电平，写 0 输出低电平。仅当端口在配置寄存器中被设为输出时，该寄存器对应位的写入值才作用于引脚；配置为输入的端口不受输出寄存器影响。读取该寄存器返回内部输出触发器的当前值，而非引脚实际电平。

极性反转寄存器：控制输入端口数据的逻辑极性。某位置 1 时，对应端口的输入数据在输入寄存器中被逻辑取反；清 0 时保留原始极性。该寄存器默认值为 0x00，即不进行极性反转。此功能仅影响输入寄存器返回的逻辑值，不改变引脚实际电平。

输入输出模式配置寄存器：设定各端口的输入/输出方向。置 1 将对应端口配置为输入，清 0 配置为输出。复位后所有位默认为 1，即全部端口上电后均为输入状态，防止上电期间引脚意外输出驱动导致总线或负载冲突。

7.5 IO 拓展芯片 I²C 读写示例

以 TPT29555A 为例，读写操作如下：

a) 单次写操作

对单个寄存器进行写入数据，时序为：

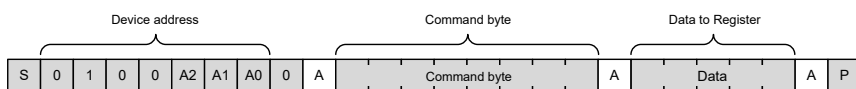


图 7-2 TPT29555A 单次写入

b) 连续写操作

对于 16-bit I/O 拓展，寄存器可以分为四类，每类中分别包含 Port0 和 Port1 两个寄存器，可以对这两个寄存器进行连续写操作。例如，在对 Port0 写入八位数据后，不发 stop 而是继续发 8 位数据，此时会继续写入相同类型的 Port1 寄存器。同理，如果是先向 Port1 写入数据，之后继续写入会将数据写入 Port0。



图 7-3 TPT29555A 连续写入

而对于 8-bit I/O 拓展，对每个寄存器进行连续写操作，寄存器地址不会递增，将重复写入同一寄存器。

c) 单次读操作

对单个寄存器进行读数据，时序为：



图 7-4 TPT29555A 单次读取

d) 连续读操作

与写操作类似，读操作也可以对每类寄存器的 port0 和 port1 寄存器进行连续读操作，时序为：

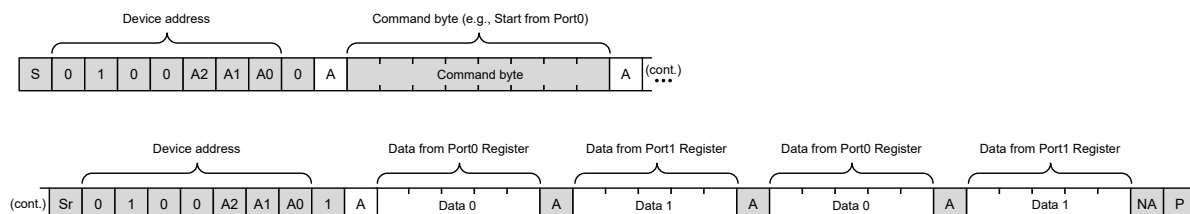


图 7-5 TPT29555A 连续读取

7.6 应用注意事项

a) IO 端口上/下拉电阻

TPT29555A 这类器件，内部集成 100k 上拉电阻，上电后默认为输入模式。如果外部没有额外下拉电阻，则输入电平为高。对于下拉的场景，可以外部

一些 I/O 拓展器件内部集成了上拉电阻。例如 TPT29555A，其端口内置 100k Ω 上拉电阻，配置为输入时引脚默认为高电平，可省去外部上拉元件。若需默认输入为低电平，可在引脚外接 1k Ω 至 10k Ω 下拉电阻将其拉低。这类集成上拉电阻的器件，可以节省外部元件和布局面积。但需注意，此类器件在输入/输出低电平时，从 V_{CC} 经过内部上拉电阻到 I/O 端口低电平形成了电流路径，会引入额外的功耗（每个 I/O 端口额外功耗为 μA 级别）。

对应低功耗应用，或者默认输入为低的应用，可以使用另一类无内部上/下拉电阻的 I/O 拓展器件，例如 TPT29535A。TPT29535A 与 TPT29555A 的引脚和寄存器完全相同，唯一区别是 TPT29535A 内部没有集成 100k 上拉电阻。其端口配置为输入时，电平完全由外部电路决定，设计者可根据需求外加独立的上拉或下拉电阻。此类器件在使用时须确保输入电平始终处于确定状态，未使用的端口不得悬空，应通过外部电阻上拉至 V_{CC} 或下拉至 GND。否则一方面容易输入电平处于不确定状态，容易引起 $\overline{INT}$ 误触发；另一方面当引脚悬空，其内部 CMOS 输入如果处于半高电平时，还会引起芯片功耗增大。对于上电后外部默认状态不确定的输入端口，同样需要外加偏置电阻以固定电平。

b) 输出模式

当 I/O 需要配置为输出时，建议优先配置输出寄存器（设置预期输出电平），再配置输入输出模式寄存器（切换引脚为输出）。若操作顺序相反，即先使能输出模式，则引脚会立即输出输出寄存器中的默认值，直至软件后续更新寄存器值。这可能导致输出存在一段时间的异常电平，对后级电路造成误触发或干扰。

8. I²C 电平转换

8.1 概述

在多电压域系统中，I²C 总线常需要连接不同供电电压的器件—例如 1.8V 的微控制器与 3.3V 的传感器、3.3V 的处理器与 5V 的 I²C 外设。若直接连接，一方面电平阈值不匹配会导致电平识别错误，另一方面低压侧器件的 I/O 引脚将承受高于其耐压限值的电平，导致可靠性下降乃至损坏。I²C 电平转换器件即为解决跨电压域 I²C 通信而设计，其核心功能是在两个不同电压总线段之间实现双向、自适应的逻辑电平转换，广泛应用于服务器、通信设备等系统。

I²C 总线中 SDA 为双向数据线，数据流向在总线事务中实时变化，地址与数据写入时为控制器到目标，应答及读取时为反方向。SCL 虽主要由控制器驱动，部分目标器件可通过时钟延展拉低 SCL，同样呈现双向行为。除了双向传输外，I²C 电平转换同时还需要满足 I²C 物理层的线与与逻辑，任意侧拉低均可使两侧总线表现为低，同时两侧低电平不能互相锁死，释放后总线可以恢复高。因此 I²C 电平转换无法通过简单的单向电平转换组合实现，必须有独特的机制来满足上述需求。

目前常见的 I²C 电平转换有多种架构方案。MOS 架构是最基础的电平转换方案，例如 TPT29306。这类芯片通过调控 MOS 栅极电压，使得低电平期间 MOS 为低阻导通，低电平可以传输到另一侧，高电平期间 MOS 关闭，实现高电平转换。由于 MOS 架构的低电平导通特性，无法实现总线电容隔离。除此之外，还有一类 I²C 中继/缓冲芯片，例如 TPT29617A。可以实现两侧信号的完全隔离和重新生成。但此类芯片必须设计特殊的方向检测机制，防止低电平锁死，同时功耗一般相比 MOS 方案更高。通过 I²C 中继/缓冲芯片可以将总线分隔成多段，增加系统总线带载能力，常应用于系统负载较重的场景或跨板连接等场景。

8.2 3PEAK I²C 电平转换选型

3PEAK 提供种类丰富的 I²C/I³C 电平转换芯片，适用于不同总线电压和场景，如下表所示。

表 8-1 3PEAK I²C/I³C 电平转换、缓冲器

特性	TPT29306	TPT29617A/X	TPCA9803	TPT29511	TPT29336	TPT29606	TPCA3BN02
功能	I ² C	I ² C 中继	I ² C 中继	I ² C 缓冲器	I ³ C 电平转换	I ³ C 电平转换	I ³ C 中继
V _{CCA} 供电	1.0V-3.6V	0.8V-5.5V	0.8V-3.6V	2.3V-5.5V	0.9V-3.6V	0.72V-1.98V	1.08V-3.6V
V _{CCB} 供电	1.65V-5.5V	2.2-5.5V	1.65V-5.5V		1.65V-5.5V	0.72V-1.98V	1.08V-3.6V
V _{CCA} ≤ V _{CCB}	✓	✓	无限制		✓	✓	无限制
使能控制	✓	✓	✓	✓	✓	✓	✓
电平转换	✓	✓	✓	不适用	✓	✓	✓
通道数量	2	2	2	2	2	2	2
最高速率	> 2 MHz	1 MHz	1MHz	400 kHz	12.5 MHz	26 MHz	12.5 MHz

此外，3PEAK 提供支持开漏应用的通用电平转换芯片，也可适用于 I²C 场景，如下表所示。

表 8-2 3PEAK 通用 OD 应用电平转换

特性	TPT20102	TPT20104	TPXF0102	TPXF0104	TPXS0102	TPXS0104
功能	OD/PP 电平转换	OD/PP 电平转换	OD/PP 电平转换	OD/PP 电平转换	OD/PP 电平转换	OD/PP 电平转换
通道数量	2	4	2	4	2	4
V _{CCA} 供电	1.4V-3.6V	1.4V-3.6V	0.8V-4.5V	0.8V-4.5V	1.1V-5.5V	1.1V-5.5V
V _{CCB} 供电	1.65V-5.5V	1.65V-5.5V	1.6V-5.5V	1.6V-5.5V	1.65V-5.5V	0.72V-1.98V
V _{CCA} ≤ V _{CCB}	✓	✓	✓	✓	✓	✓
使能控制	✓	✓	✓	✓	✓	✓
电平转换	✓	✓	✓	✓	✓	✓
最高速率	1.2Mbps OD 100Mbps PP	1.2Mbps OD 100Mbps PP	100Mbps	100Mbps	2Mbps OD 24Mbps PP	2Mbps OD 24Mbps PP

8.3 分立 MOS 电平转换电路

8.3.1 电路结构和工作原理

利用分立 MOS 搭建电平转换电路，可以实现双向电平转换，电路结构如下所示。每路通道由一个 N 沟道 MOSFET 和 2 个上拉电阻构成。MOS 的栅极接至低压侧总线电源 V_{CC1} ，源极接低压侧总线信号，漏极接高压侧总线信号。低压侧与高压侧总线通过电阻上拉至各自的电源轨。I²C 总线电平转换，SDA 和 SCL 共需两个 MOS 实现。

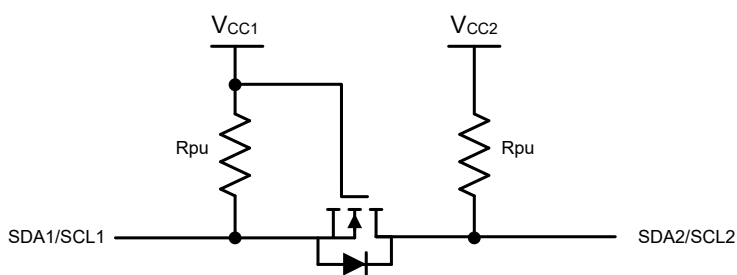


图 8-1 分立 MOS 电平转换电路

该电路工作依赖于 N 沟道 MOS 的开启特性：仅当栅极-源极电压差大于 MOS 开启阈值电压 V_{TH} 时，MOS 导通。由于 MOS 栅极固定接 V_{CC1} ，因此 MOS 导通与否取决于源极输入电压。

低电平传输：

当低压侧器件输出低电平时，源极电压趋近 0 V， $V_G - V_S \approx V_{CC1} > V_{TH}$ ，MOS 沟道导通，因此高压侧总线被同步拉低。当高压侧器件输出低电平时，首先漏极降至接近 0 V，由于 MOS 存在体二极管，因此源极也随之被拉低，最终 MOS 导通。因此无论哪一侧发起低电平，均可传向另一侧，实现双向传输和线与功能。

高电平传输：

当两侧器件均释放总线后，在上拉电阻作用下总线电压上升。总线电压上升至 $V_{CC1} - V_{TH}$ 时，MOS 关断。随后两侧总线被各自电阻上拉。高电平电压完全由对应侧 V_{CC} 决定，实现电平转换。

上述机制正常工作的前提是 V_{CC1} 须高于 MOS 的 V_{TH} ，确保低电平期间 MOS 可以导通。

8.3.2 优势与劣势

该方案的主要优势在于结构简单、成本较低。但分立 MOS 方案存在以下固有限制：

- NMOS 存在寄生体二极管，如果低压侧先上电，将通过体二极管向高压侧漏电。因此该方案对上下电时序有严格要求。

- 总线电平需要受限于 MOS 阈值电压，对于超低电压 I²C 总线，适配的 MOS 型号较少。
- 延时和速率受到 MOS 限制，且多路一致性较差。
- 未集成 ESD 等保护器件，抗干扰能力较差。

随着系统设计向小型化、低功耗和高速率方向演进，分立 MOS 电平转换方案在诸多应用中已难以满足要求。集成式开漏电平转换器件在保持该架构灵活性的同时，以更小的封装、更低的功耗和确切的时序参数逐步成为主流替代方案，详见[集成 MOS 架构电平转换器](#)节。

8.4 集成 MOS 架构电平转换器

8.4.1 概述

分立 MOS 搭建电平转换电路虽结构简单，但其固有的阈值电压约束、体二极管漏电以及占板面积大等问题，限制了其在低电压、高速率和小型化系统中的进一步应用。集成 MOS 架构电平转换器正是在分立方案基础上优化演进而来的单片集成方案，将多个 MOS 开关通道及偏置控制电路集成于单一芯片内，在保留外部上拉电阻灵活性的同时，解决了分立方案的诸多痛点。3PEAK 的此类器件的典型代表包括 TPT29306、TPT29336、TPXF0102、TPXF0204 等，均为集成 MOS 方案架构。

8.4.2 工作原理

集成 MOS 电平转换器的核心仍为 N 沟道 MOSFET，但其 MOSFET 结构与栅极偏置方式与普通 MOS 存在差异。以 TPT29306 为例，其内部集成包含 SCL，SDA，V_{REF} 三个 MOS 通道。这几个 MOS 的栅极短接在一起，对外为 EN 引脚。SCL 和 SDA 通道用来进行总线电平转换，V_{REF} 通道的 MOS 通过特定接法来产生合适的栅极偏置电压。

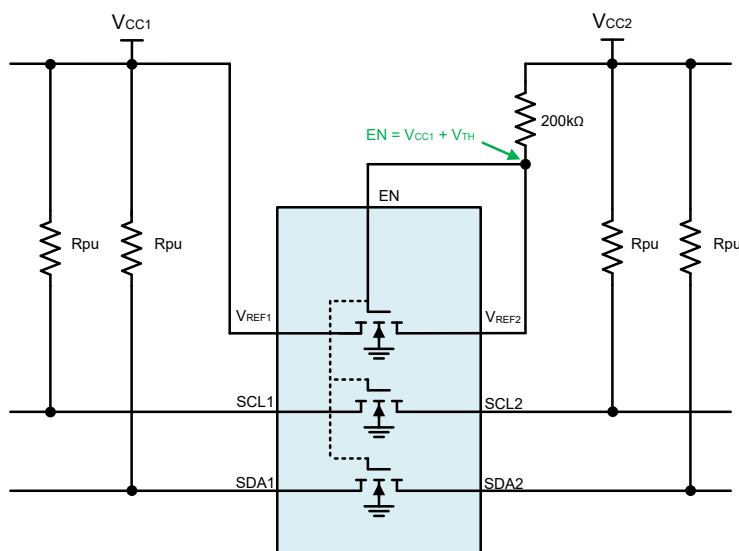


图 8-2 TPT29306 集成 MOS 电平转换架构

典型应用中，V_{REF1} 接低压侧电源 V_{CC1}，而 V_{REF2} 则需要在外与 EN 短接，再经过偏置电阻（一般为 200kΩ）上拉到高压侧电源 V_{CC2}。这种接法将形成负反馈，可以将 MOS 的栅极电压 EN 自动调节在 V_{REF1}+V_{TH}，其中 V_{TH} 为 MOS 的开启阈值。此时 SDA 与 SCL 通道的栅极电压也将同步为 V_{REF1}+V_{TH}。其电平转换机制如下：

低电平导通：当任一侧信号被拉低到 V_{REF1} 以下时，MOSFET 栅极电压与输入信号压差将大于 V_{TH}（V_{EN} - V_{IN} > V_{TH}），MOSFET 处于导通态。因此低电平可以直接传输到另一侧，将另一侧总线拉低。

高电平关断：当两侧器件均释放总线后，总线电压被电阻上拉。随着电压上升，栅极电压与输入信号压差逐渐减小，MOS 内阻增大。当总线电压上升至 V_{REF1} 时，此时 MOSFET 内阻已经增大至关断状态。因此两侧总线被隔离，高压侧总线将继续被电阻上拉到 V_{CC2}，实现电平转换。

另外，该系列芯片内部的 MOSFET 与分立 MOSFET 器件相比，做了一些特殊优化。N 沟道 MOSFET 结构如图 8-3 所示，其中源极（S）和漏极（D）为 N 型掺杂，衬底 B 为 P 型掺杂，因此从 B 到 S 和 D 形成两个 PN 节（二极管）结构。我们常见的分立 NMOS 器件，其衬底 B 一般与源极 S 短接，因此其源极（衬底）到漏极形成体二极管结构，如图 8-4 所示，因此天然存在从 S 极向 D 极的漏电或倒灌路径。

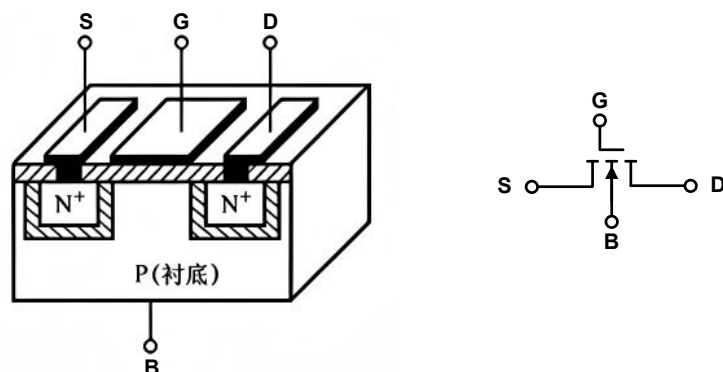


图 8-3 NMOS 内部结构图

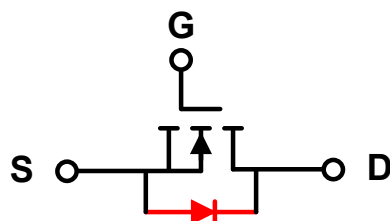


图 8-4 分立 NMOS 器件单向体二极管

而集成 MOS 电平转换器件内部的 MOSFET，其衬底没有与源极短接，而是接 GND。因此源极（S）和漏极（D）之间不是单向体二极管，而是为一对方向相反的背靠背二极管，因此不会像分立 MOSFET 那样形成单向漏电路径。此外，这种结构下源极和漏极完全对称的，任意一侧电压为低电平，MOSFET 均可以导通，从而实现自动双向电平传输。

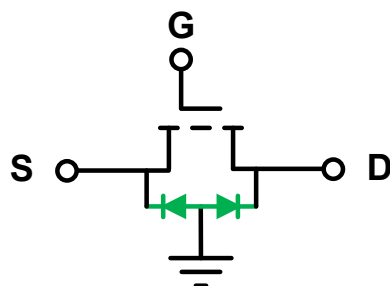


图 8-5 集成 MOS 器电平转换器件双二极管

8.4.3 典型应用电路

从这类芯片的工作机制可以看出，当 EN 与 V_{REF2} 短接并上拉到 V_{CC2} 时，电平转换功能开启；而当 EN = 0V 时，整个输入信号范围内 MOSFET 均不满足开启条件，通道处于关断高阻态。通过这种方法控制 EN 电压，可以实现通道使能控制。

芯片的 EN 实际上为 MOSFET 的栅极电压，因此不允许直接通过推挽信号去控制 EN。常见的做法有两种，一种是依旧将 EN 和 V_{REF2} 短接，再经过偏置电阻后接到推挽信号控制。另一种是使用开漏信号对 EN 进行控制。以 TPT29306 为例，其常见应用电路如下：

a) 无使能控制电路，通道保持开启

这个是最常见的应用电路。其中 EN 与 V_{REF2} 短接后，通过偏置电阻（一般为 200kΩ）上拉到高压侧电源。此外建议在 EN 引脚增加 100pF~100nF 的电容，保证栅极电压的稳定。

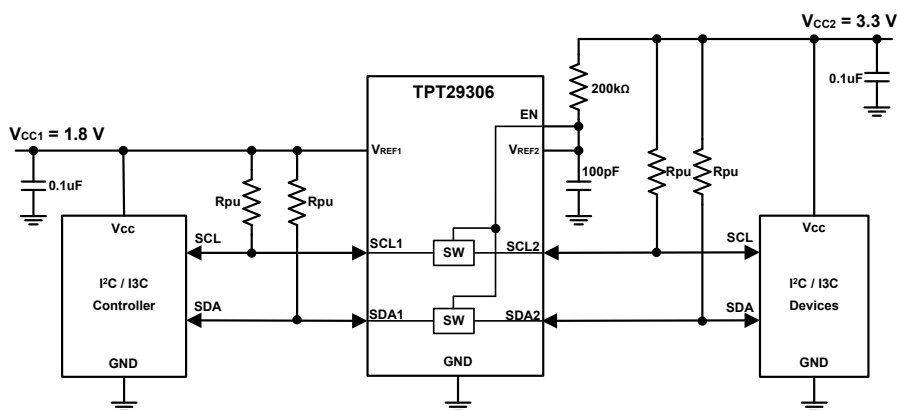


图 8-6 TPT29306 典型应用电路

b) 推挽信号使能控制

如果需要推挽信号进行通道使能控制，可以将 EN 与 V_{REF2} 短接后经过偏置电阻后接到推挽信号控制。当控制信号为高时，EN 电压可以正常偏置，实现通道开启。当控制信号为低（0V）时，EN 电压将稳定在 0V，通道 MOS 将保持关断。

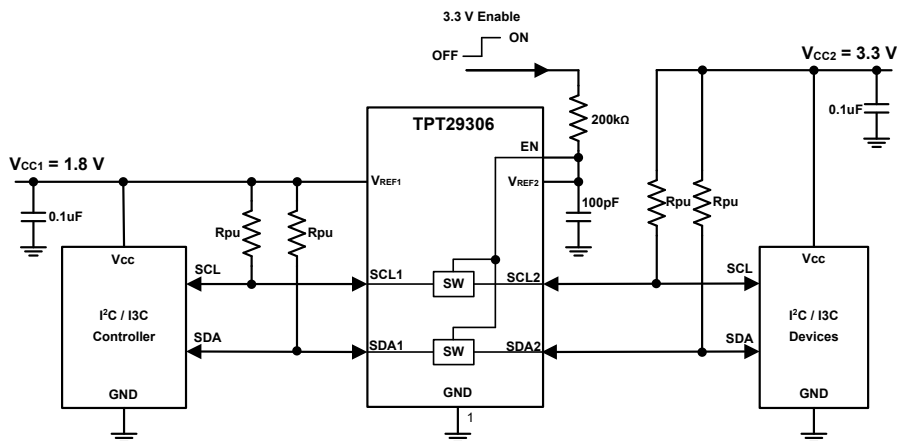


图 8-7 TPT29306 推挽信号使能控制电路

c) 开漏信号使能控制

另外一种方案是使用开漏信号进行控制。当开漏输出高时，EN 与 V_{REF} 保持正常偏置状态。当开漏输出低时，EN 被直接施加 0V，此时所有通道 MOSFET 也处于关断状态，实现使能控制。

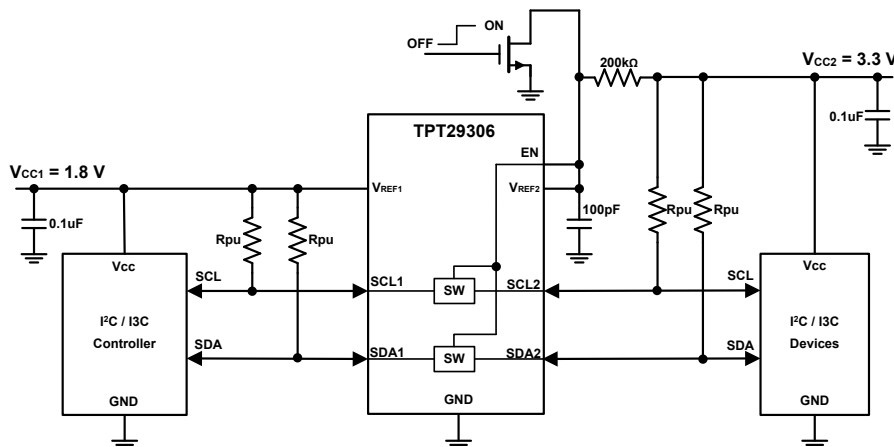


图 8-8 TPT29306 开漏信号使能控制电路

8.4.4 EN 和 V_{REF2} 的常见错误接法

TPT29306 与 TPXF010x 系列的 EN 引脚接法是器件正常工作的关键，接法不当会导致器件工作异常。最常见一些接法错误以及如下：

a) EN 直接接高压侧电源 VCC2

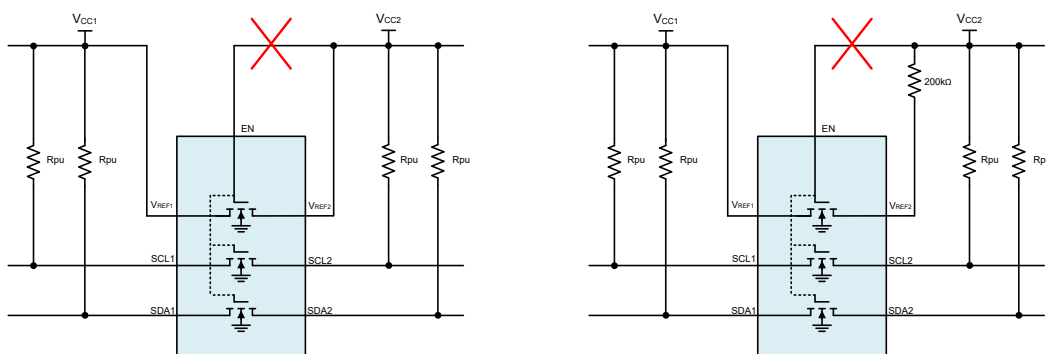


图 8-9 错误接法：EN 直接接高压侧电源 V_{CC2}

影响：这是最常见的错误接法。这种接法下，MOSFET 栅极电压固定为 V_{CC2} ，如果 $V_{CC2} > V_{CC1} + V_{TH}$ ，一方面 V_{REF} 路径的 MOSFET 将低阻导通， V_{REF1} 与 V_{REF2} 可能流过太大电流造成破坏。另一方面，1 侧的高电平输入为 V_{CC1} ，低于 $EN - V_{TH}$ ，因此高电平期间 MOS 无法关断，1 侧高电平将被 2 侧总线抬高，直到 MOS 关闭。最终导致 1 侧总线电压将被抬升到 V_{CC1} 以上，这可能导致通讯异常，严重时甚至造成低压侧器件损坏。

正确的电路接法应如图 8-6 所示。

b) EN 直接接推挽信号

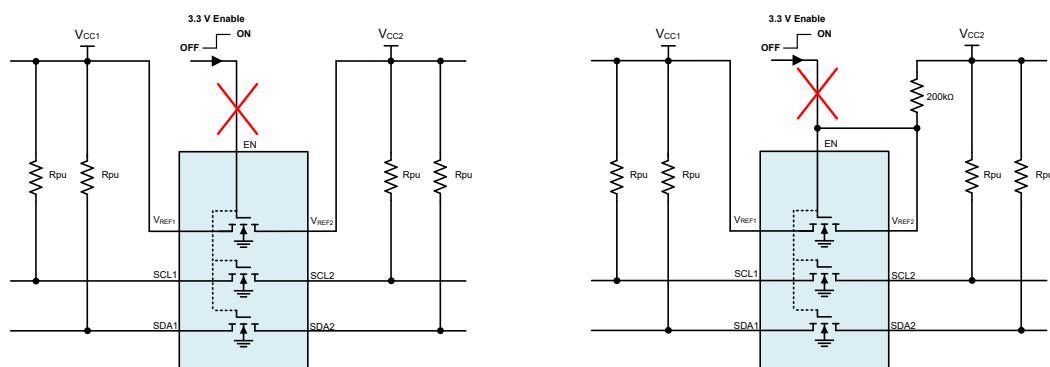
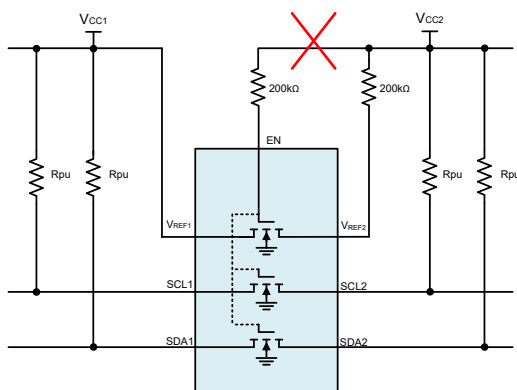


图 8-10 错误接法：EN 直接接推挽信号

影响：这种接法下，当推挽信号输出为高时，如果此电平高于 $V_{CC1} + V_{TH}$ ，那同样将导致高电平无法关断。与 EN 直接接高压侧电源 V_{CC2} 效果类似。

实际应用中，如果需要推挽信号控制，应按照图 8-7 所示电路。

c) EN 电阻上拉，但没有与 V_{REF2} 短接


图 8-11 错误接法：EN 电阻上拉，但没有与 V_{REF2} 短接

影响：这种接法下，EN 与 V_{REF2} 独立上拉到 V_{CC2} ， V_{REF} 通道 MOS 无法形成负反馈去自动调节 EN 电压。MOS 栅极电压 EN 电压最终等于 V_{CC2} ，因此器件同样无法正常工作。

8.4.5 功耗

这类 MOS 架构电平转换，功耗非常低。其静态电流实际为 $200k\Omega$ 电阻上流过的电流，从 V_{CC2} 经过 $200k\Omega$ 电阻以及 MOS 流向 V_{CC1} 。如下图所示：

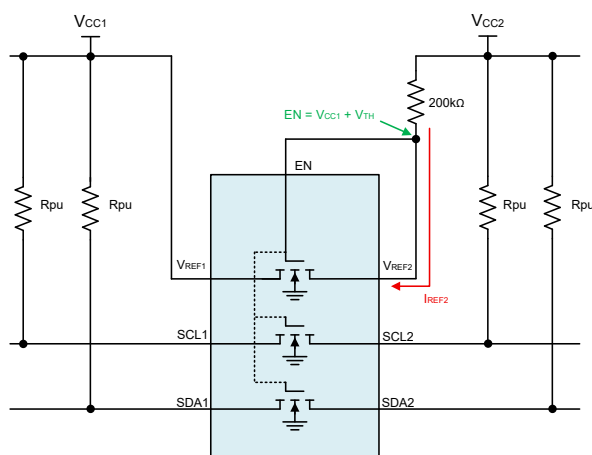


图 8-12 功耗

计算公式为：

$$I_{CC2} = (V_{CC2} - V_{CC1} - V_{TH})/200k\Omega$$

其中， V_{TH} 为 MOS 的导通的阈值电压，TPT29306 约为 0.6V。例如，当 $V_{CC1} = 1.8V$, $V_{CC2} = 3.3V$ 时，计算得 $I_{CC2} = 4.5\mu A$ 。

对于 V_{CC1} 而言，电流方向为流入。在一些 V_{CC1} 采用 LDO 供电的场合，如果该 LDO 无法接受灌电流且没有其他负载，则需要将 V_{CC1} 对 GND 接弱下拉电阻来提供电流路径，下拉电阻大小需要满足： $R_{pull} < V_{CC1}/I_{CC2}$ 。

8.4.6 上拉电阻

这类器件为无源 MOS 电平转换器件，高电平依赖外部上拉电阻建立。上拉电阻的取值影响信号边沿速率、功耗以及电平转换的可靠性。两侧上拉电阻最大值，需要根据各自总线电容和 I²C 速率决定，其计算方法可以参考 I²C 上拉电阻选取节。需要注意的是，上拉电阻最小值，受到总线其他器件的 I_{OL} 限制。应根据总线上器件的 I_{OL} （低电平吸电流）能力，选取合适的上拉电阻，防止上拉电阻过小导致低电平抬升。

表 8-3 列出了系统 I_{OL} 电流为 8mA、5mA 和 3mA 时的上拉电阻最小值。

表 8-3 上拉电阻最小值限制

上拉电压 (V)	8 mA		5 mA		3 mA	
	典型值 (Ω)	+10% (Ω)	典型值(Ω)	+10% (Ω)	典型值(Ω)	+10% (Ω)
5 V	581	639	930	1023	1550	1705
3.3 V	369	406	590	649	983	1082
2.5 V	269	296	430	473	717	788
1.8 V	181	199	290	319	483	532
1.5 V	144	158	230	253	383	422
1.2 V	106	117	170	187	283	312

在一些特殊应用场景下可以省略上拉电阻，以 TPXF0102 为例，上拉电阻需求如下：

a) A 侧（低压侧）向 B 侧（高压侧）电平转换

当信号从 A 侧驱动至 B 侧且 An 端口为高电平时，MOS 关闭，Bn 端口通过连接至上拉电源电压（ V_{PU} ）的上拉电阻被驱动至高压侧总线电压。这种情况下 B 侧（高压侧）始终需要上拉电阻，A 侧（低压侧）则在前级输出为开漏结构或输入漏电流超过 $1\mu A$ 时，也需要上拉电阻。

b) B 侧（高压侧）向 A 侧（低压侧）电平转换

当信号从 Bn 端口驱动至 An 端口为高电平时，MOS 关闭，可以将 An 端口（如果未加上拉）的电压钳位到由 V_{REFA} 设定的电压。如果 B 侧（高压侧）信号始终由推挽驱动，可省略 B 侧的上拉电阻；若 A 侧（低压侧）接收端的漏电流小于 $1\mu A$ ，则也可省略 A 侧的电阻，通过器件本身将 A 侧电压钳位到 V_{REFA} 。而对于开漏类型的驱动器，必须保留 B 侧的上拉电阻，因为开漏输出本身无法自行驱动高电平。

8.4.7 推挽电平转换

MOS 架构电平转换芯片除了支持 I²C 等开漏电平转换外，还可以支持推挽电平转换。以 TPXF0102 为例，推挽电平转换应用如下：

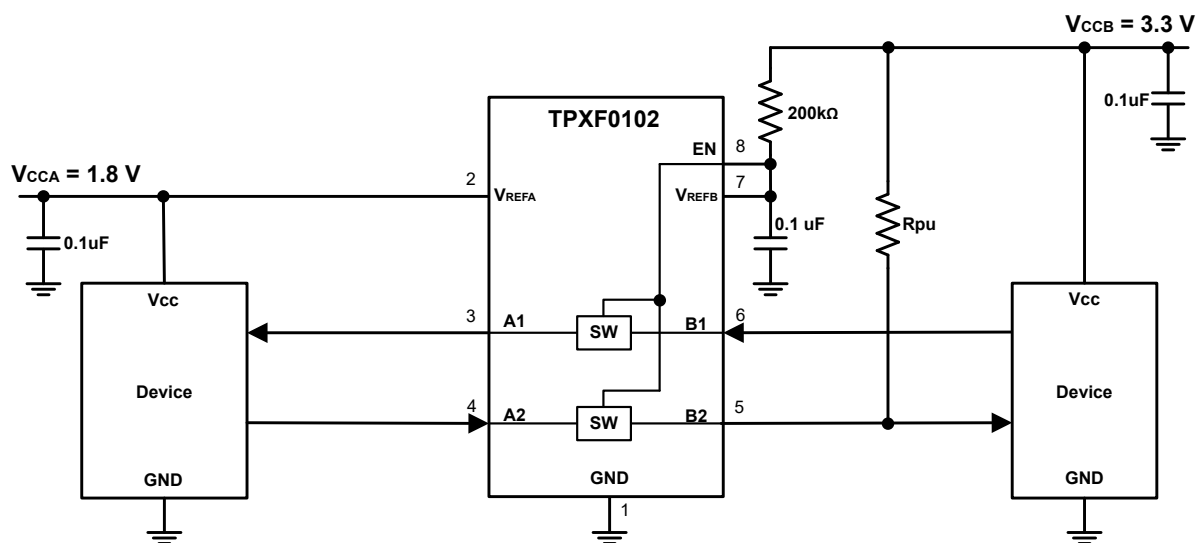


图 8-13 推挽电平转换

a) 高压侧向低压侧推挽电平转换

对于上升沿，高压侧驱动电平上升期间，由于 MOS 在 V_{REFA} 以下导通，因此低压侧总线也可以被快速上拉。对于下降沿，当电压下降至 V_{REFA} 之下时，MOS 开启，低压侧总线电平也被快速下拉，从而实现推挽电平转换。MOS 开启期间 MOS 内阻很小，因此可以实现快速边沿变化和极低的传播延时。

高压侧向低压侧推挽电平转换典型波形如下图所示。

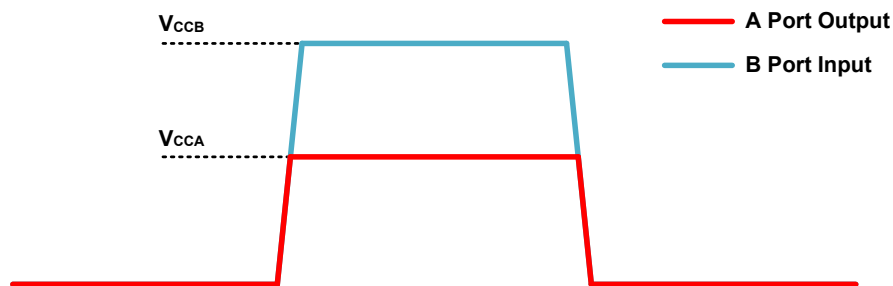


图 8-14 高压侧向低压侧推挽电平转换波形

b) 低压侧向高压侧推挽电平转换

低压侧向高压侧推挽电平转换时，上升沿 B 侧输出的可以分为两段。第一段为信号从低电平上升到 V_{CCA} 期间，MOS 导通，因此 A 侧推挽信号可以直接传输到 B 侧，上升沿速率很快。第二段为当电平上升到 V_{CCA} 之后，MOS 关断，因此之后 B 侧总线电压只能被上拉电阻拉高，第二段速率主要受到高压侧上拉电阻和高压侧总线电容限制。对于需要高速率上行推挽传输应用，需要把电平转换器件尽可能靠近高压侧器件，从而减小高压侧寄生电容，提高上升沿后半段速率。

对于下降沿，当 A 侧电压下降到 V_{CCA} 以下时，MOS 开启，因此 B 侧信号可以被快速下拉。

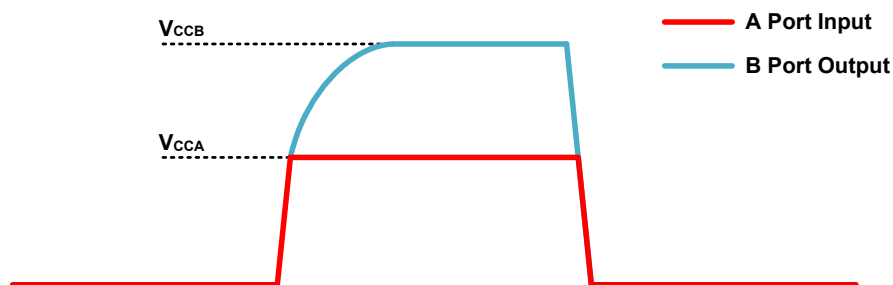


图 8-15 低压侧向高压侧推挽电平转换波形

8.4.8 集成 MOS 器件相比分立方案 MOS 优势

相比分立 MOS 方案，这类集成 MOS 电平转换芯片优势如下：

特性	集成 MOS 电平转换	分立 MOS 电平转换
方案复杂度	低	高
元件数	低	高
功率耗散/位	低	高
已知上升/下降时间	是	否
数据速率支持	是	受限
信号干扰	否	有可能
需要外部 ESD	否	是

特性	集成 MOS 电平转换	分立 MOS 电平转换
需要电源时序	否	是

8.4.9 应用注意事项

a) 最高通讯速率

TPT29306, TPXF0102 此类架构的器件, 最大通讯速率与总线电容以及上拉电阻大小相关。总线电容较小且上拉电阻合适的情况下, TPT29336 可以支持 >12.5MHz 速率的通讯, TPXF0102 可以支持 >40MHz 速率的通讯。一般总线电容固定的情况下, 上拉电阻越小, 通讯速率越高。但上拉电阻减小时, 器件输出低电平时的 I_{OL} 会增大, 导致低电平出现抬升。实际应用中需要根据寄生电容大小和速率需求选择合适的上拉电阻, 对于高速率应用, 应尽可能减小总线电容负载。

b) 与中继器/缓冲器的区别

需注意 TPT29306 和 TPXF010x 这种 MOS 架构的电平转换, 不具备两侧电容隔离和信号重驱动能力。MOSFET 导通时两侧总线电容直接并联, 因此总电容叠加后须仍满足 I²C 协议 C_b 上限要求。若系统总电容负载较重, 应采用 I²C 中继/缓冲器进行电容隔离。

c) TPXF020x 与 TPXF010x 的区别

TPXF020x 与 TPXF010x 采用相同 MOS 架构, 但 TPXF020X 系列可以支持 EN 推挽信号控制。其外部偏置电阻集成在了芯片内部, 用户可以将 V_{REF2} 直接接 V_{CC2} 。另外, EN 为数字输入引脚, 经过内部电平调理后再对栅极电压进行控制, 因此 EN 可以直接接推挽信号进行控制, 简化了控制方式。

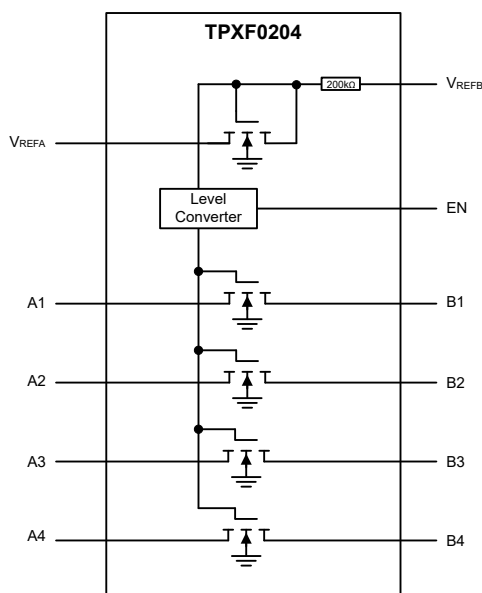


图 8-16 TPXF0204 结构框图

d) VCC1 = VCC2 应用

在一些场景下 A、B 侧电压相同, 芯片仅做开关来使用, 电路依旧可以按照之前的接法。两侧总线电压均为 V_{CC} 时, 内部 MOSFET 的栅极的 EN 电压为 V_{CC} 。当输入信号低于 $V_{CC}-V_{TH}$ 时, MOS 导通, 因此只要低电平

幅值低于该值，依旧可以正常传输。当信号上升到 $V_{CC}-V_{TH}$ 时，MOS 关闭，之后两侧总线被各自的上拉电阻上拉到 V_{CC} 。这种应用下，无论是开漏电平或推挽电平转换，两侧必须加外部上拉电阻，否则输出高电平只能达到 $V_{CC}-V_{TH}$ ，无法达到正确高电平电压。

8.5 带边沿加速的 MOS 架构电平转换器

8.5.1 概述

MOS 架构的电平转换芯片速率受到上拉电阻和总线电容限制，如果要提高速率，则需要减小上拉电阻，同时会导致低电平抬升和功耗增加。在此基础上，衍生出一类带边沿加速电路的电平转换芯片，在使用弱上拉电阻的同时，采用边沿加速电路来提高速率，从而实现高速率和低功耗。例如 3PEAK 的 TPT29606，TPT20102，TPXS0102，TPCA3BN02 等芯片。

这类芯片在 MOS 架构基础上集成了单稳态（One-Shot）边沿加速电路，在信号上升或下降沿期间提供短暂强上/下拉，显著提升边沿速率。其内部一般集成弱上拉电阻或高电平保持器（High-keeper），一般可以省略外部上拉电阻。减少了外部元件数量，在保持低功耗的同时实现更高的速率。这类通用电平转换芯片可以支持开漏信号和推挽信号电平传输，可以用来做 I²C 电平转换，并且可以支持 I3C，SPI 等更高速率的应用。

8.5.2 工作原理

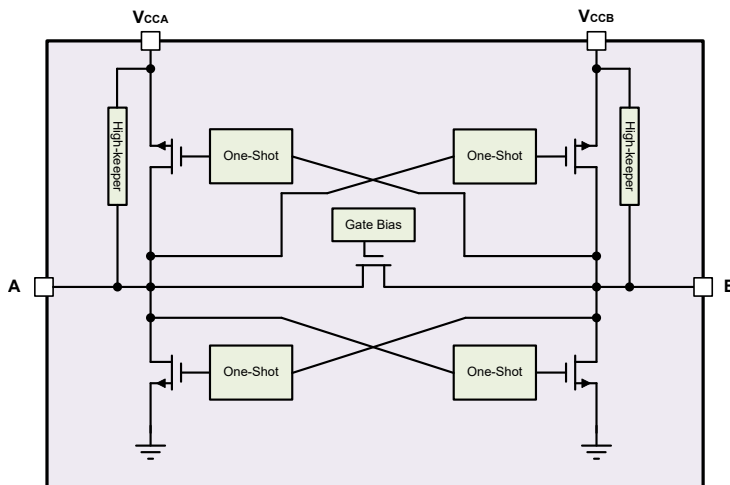


图 8-17 集成边沿加速功能的 MOS 电平转换芯片架构示意图

该类器件除了横置 MOS 外，还在通道上增加了对 V_{CC} 或 GND 的推挽驱动管，其只在边沿加速期间开启。在稳态期间，其工作原理与上一节 MOS 架构电平转换类似，低电平期间 MOS 导通，电平传输到另一侧。高电平期间 MOS 关闭，两侧总线被各自电阻上拉，实现电平转换。值得注意的是，该类器件内部一般集成上拉电阻，根据型号差异，上拉电阻值和其开启或禁用条件也可能不同，具体以数据手册为准。

当检测到上升沿或下降沿时，会触发内部的单稳态电路，产生 10~30ns 的脉冲信号（不同器件的单稳态脉冲宽度也可能不同），打开相应的边沿加速 MOS，使信号快速上升/下降到总线高/低电平。部分器件只存在上升沿加速电路，下降沿通过低电平时横置 MOS 开启实现将另一侧下拉。

这类器件同时支持开漏和推挽电平转换。对于开漏信号，当总线被释放为高电平时，首先被内部上拉电阻缓慢上拉，当波形上升到加速阈值点，加速 MOS 开启，波形被快速上拉。开漏信号传输最高速率主要受到前半段电阻上拉时间限制，与总线负载电容和上拉电阻大小有关。下降沿时输出侧首先被加速下拉，下降到低电平后横置 MOS 开启，继续保持低电平输出。开漏信号典型波形如下图所示。

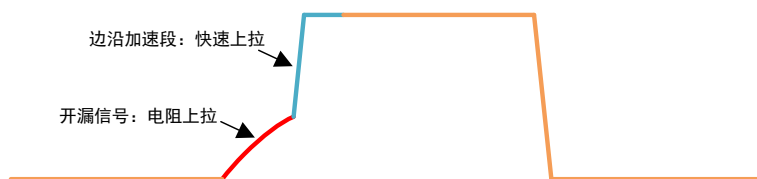


图 8-18 开漏信号波形

对于推挽信号，由于其输入信号边沿本身较快，因此可以快速触发加速电路，从而达到更高的速率。输入推挽信号的脉冲宽度不能低于加速时间，否则输入电平变化时，加速 MOS 还未关闭，会与输入信号互相竞争。因此推挽信号最高速率主要取决于单稳态加速时间限制，例如 15ns 的加速脉冲时间，对应最高的推挽信号速率不能超过 66Mbps。开漏信号典型波形如下图所示。



图 8-19 推挽信号波形

这类器件边沿加速结束后，高电平通过内部上拉电阻维持，低电平通过横置 MOS 维持。因此其稳态驱动能力受到上拉电阻阻值和 MOS 内阻限制。其上拉电阻一般为弱上拉，因此应用中需要尽量避免外部存在下拉电阻的场景，防止高电平被拉低。外部增加额外上拉电阻时，需要考虑 MOS 内阻，如果外加较小上拉电阻，会导致输出低电平发生抬升。

8.5.3 应用注意事项

a) 外部上拉电阻

这类带边沿加速的器件内部一般都集成上拉电阻，可以在稳态期间提供高电平输出。部分型号在低电平和高电平期间均存在上拉，例如 TPCA3BN02、TPT20102，这类从低电平上升到高电平时，达到边沿加速阈值之前可以通过内部上拉电阻产生上升沿，因此外部可以省略上拉电阻。另外 TPT29606，则低电平期间会自动禁用内部上拉电阻，这颗在用来做 I²C 电平转换时，需要外部加上拉电阻。否则外部低电平释放后，没有上拉路径来生成上升沿信号。考虑到兼容不同器件需求，建议在 I²C 应用中，均预留上拉电阻位置，根据手册要求进行上拉电阻配置。

b) 长走线应用

这类器件的上升/下降沿速率很快，等效为高频信号。当走线比较长且阻抗不匹配或不连续时，信号反射可能引起振铃和回勾现象。最恶劣的情况下，振铃可能引起单稳态电路误触发，导致边沿出现 Glitch 或者振荡。应用中需要注意，走线长度尽可能足够短，以便任一信号反射的往返延迟低于器件本身的单稳态时间。对于长走线应用，建议加串阻来抑制振铃，优化信号质量。

8.6 I²C 中继器

8.6.1 概述

I²C 中继器（Repeater）是一类对 I²C 总线信号进行重驱动和电容隔离的器件。与无源 MOS 电平转换方案不同，中继器内部集成有源驱动电路，能够在低电平期间主动下拉输出侧，在高电平期间释放总线并由外部上拉电阻完成高电平建立，从而将两侧总线的电容负载彼此隔离。这一特性使得中继器特别适用于长距离 I²C 通信、大型背板中多板卡总线互联，以及负载电容超出 400pF 协议限制的复杂系统。

TPT29617A 是一款支持电平转换功能的双向 I²C 中继器，支持 SDA 和 SCL 两路信号的同时重驱动。器件内部集成自动方向检测逻辑和独特的低电平弱驱动机制，可在不引入总线锁死风险的前提下实现两侧电容隔离，最高可以支持 1MHz I²C（快速模式增强模式）速率，支持跨电压域双向通信。

8.6.2 工作机制

与 MOS 架构电平转换不同，I²C 中继需要实现双向信号传输的同时，避免两侧低电平驱动电路形成正反馈锁死回路。锁死的原因如下：

I²C Repeater 器件两侧信号为重新驱动生成。假设无特殊防锁死机制，当 1 侧为低电平时，逻辑控制 MOS2 打开，使 2 侧输出低电平；而 2 侧的低电平又会反向导致 MOS1 打开，使 1 侧输出低电平。两侧 MOS 互相保持对方的开启条件，即使外部器件释放总线，低电平状态仍被持续锁定。总线因此永久瘫痪，通信无法恢复。

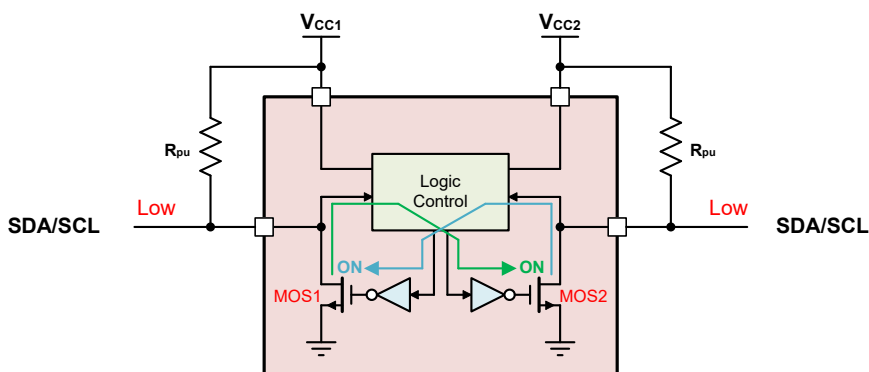


图 8-20 锁死机制

TPT29617A 这类 I²C 中继芯片，通过其特殊的方向检测机制，可以防止总线锁死。TPT29617A 的 V_{OL} 和 V_{IL} 如下表所示：

表 8-4 TPT29617A V_{OL} 和 V_{IL} 特性

参数	端口	最小值	典型值	最大值	单位
V_{OL}	SCL1, SDA1		0.1	0.2	V
	SCL2, SDA2		0.5	0.6	V
V_{IL}	SCL1, SDA1			30% V_{REF1}	V
	SCL2, SDA2			0.4	V

TPT29617A 通过将 2 侧 V_{OL} 设置高于其自身 V_{IL} ，来防止锁死。其中 V_{OL} 为 0.5V，该电压足以被外部目标器件识别为低电平（I²C 规范低电平阈值为 30% VCC，2 侧最低供电电压为 2.2 V，对应外部器件 V_{IL} 为 0.66 V），该电平高于 TPT29617 自身 2 侧的 V_{IL} 0.4 V，因此不会发生图 8-20 的锁死现象。而 TPT29617A 当接收外部器件输出的低电平时，根据 I²C 规范外部器件的 V_{OL} 应低于 0.4V，因此 TPT29617A 也能正常识别。

a) TPT29617A 1 侧向 2 侧电平传输过程

控制器驱动 SDA1 为低电平时，MOS2 打开，SDA2 被内部弱驱动拉至 0.5 V。该电压高于 2 侧 V_{IL} ，因此 MOS1 保持关闭。如图 8-21 所示。

控制器释放总线后，SDA1 通过电阻上拉回升至高电平，MOS2 随即关闭，SDA2 也通过电阻上拉恢复高电平。

SCL 与 SDA 架构相同，也为相同的表现。

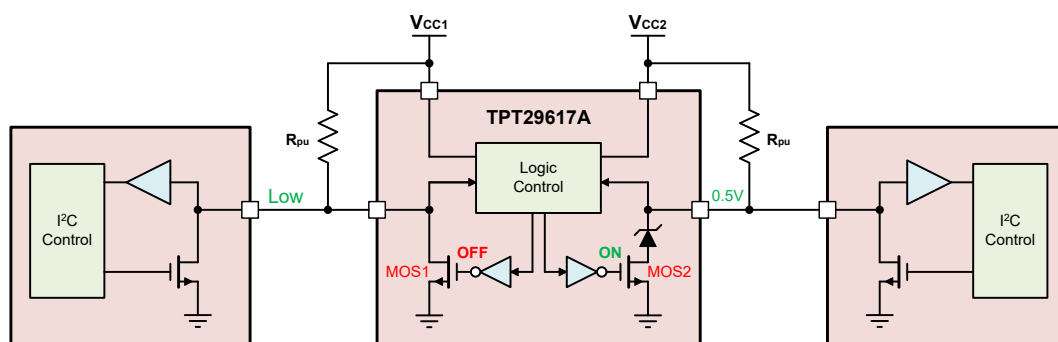


图 8-21 TPT29617A 1 侧向 2 侧电平传输过程

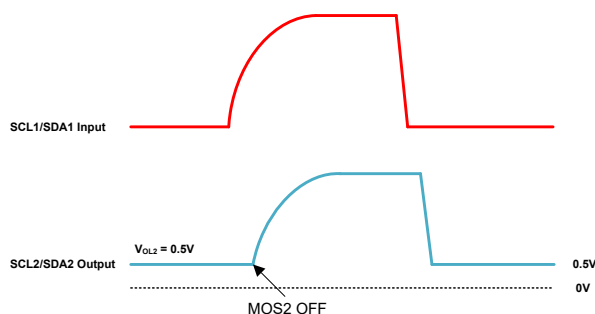


图 8-22 TPT29617A 1 侧向 2 侧电平传输波形

b) TPT29617A 2 侧向 1 侧电平传输过程

当目标器件驱动 SDA2 为低电平（低于 0.4 V）时，MOS1 打开，SDA1 被下拉至低电平。此时 1 侧检测到低电平输入时，MOS2 也受控打开，由于其 0.5 V 为弱驱动，驱动能力远逊于目标器件，因此 SDA2 可以继续被目标器件驱动至 0.4 V 以下。如图 8-23 所示。

目标器件释放总线后，SDA2 首先被内部弱驱动拉至 0.5 V，使 MOS1 关闭。SDA1 电压开始上升，在上升到高电平之前，1 侧仍检测到低电平，因此 MOS2 保持开启，SDA2 将在 0.5V 维持一段时间。当 SDA1 上升高电平后，MOS2 关闭，SDA2 也被释放回升至高电平。这个过程典型波形如图 8-24 所示。当 SDA1 总线电容负载越重或上拉电阻越大，那么 SDA2 侧 0.5V 台阶停留的时间越长。

SCL 与 SDA 架构相同，也为相同的表现。

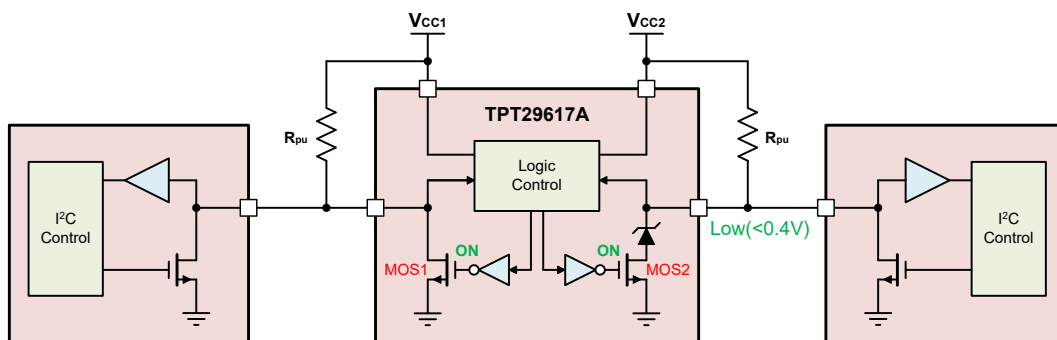


图 8-23 TPT29617A 2 侧向 1 侧电平传输过程

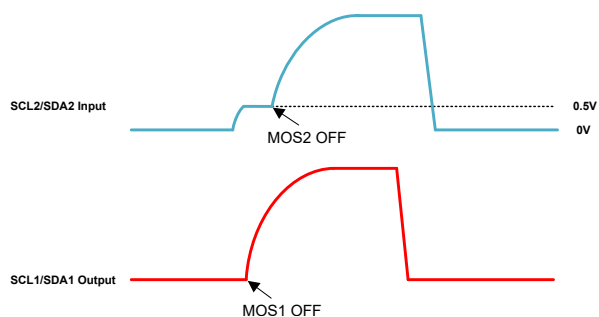


图 8-24 TPT29617A 2 侧向 1 侧电平传输波形

8.6.3 TPT29617A 典型 I²C 应用波形

TPT29617A 的 1 侧和 2 侧任意一侧均可连接 I²C 控制器或目标器件。其典型通讯波形如下所示。

a) 1 侧为控制器，2 侧为目标器件

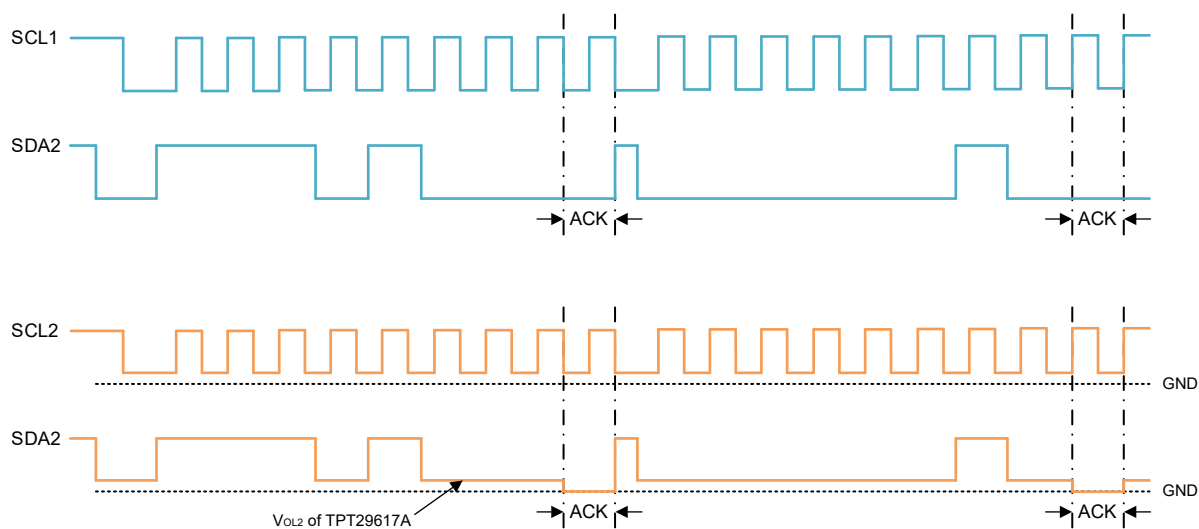


图 8-25 1 侧为控制器，2 侧为目标器件时总线波形

当 1 侧为控制器，2 侧为目标器件时，对于时钟信号以及来自控制器发送的数据，2 侧低电平输出为 0.5V。而在 ACK 以及读取数据时，2 侧总线由目标器件控制，此时低电平被正常拉低到 0.4V 以下。从波形可以看到对应的低电平变化过程。

b) 2 侧为控制器，1 侧为目标器件

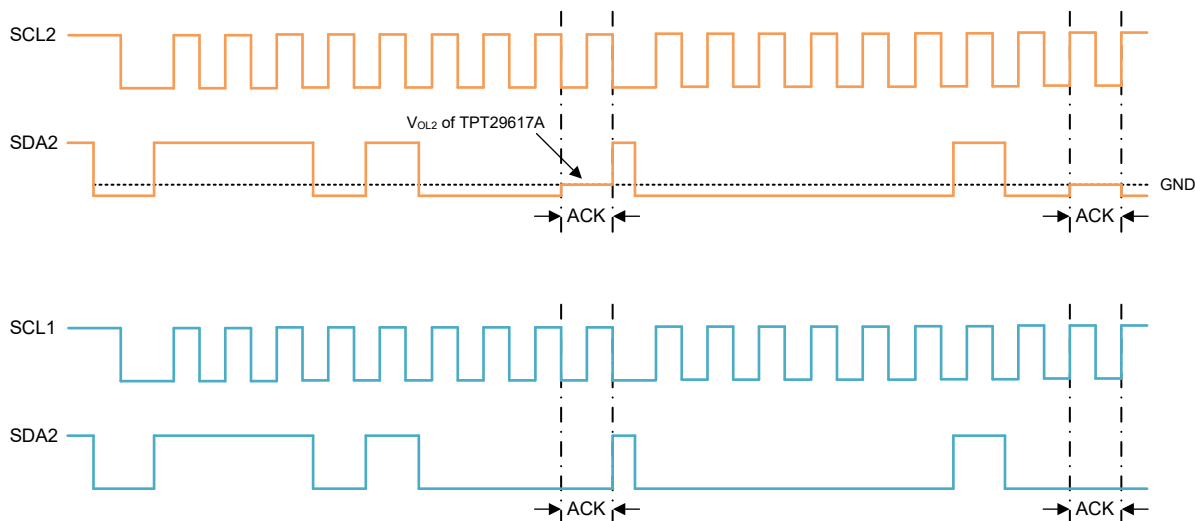


图 8-26 2 侧为控制器，1 侧为目标器件时总线波形

当 2 侧为控制器，1 侧为目标器件时，对于时钟信号以及来自控制器发送的数据，2 侧低电平被正常拉低到 0.4V 以下。而在 ACK 以及读取数据时，电平由 1 侧向 2 侧传输，此时也可以在 2 侧总线看到低电平抬升到 0.5V。

8.6.4 应用注意事项

a) 中继器与 MOS 架构电平转换器的适用场景

TPT29617A 为 I²C 总线中继器，信号为重新驱动输出，可以隔离前后总线的电容。常用于长走线，重电容负载，跨板通讯，信号隔离保护等场景。

TPT29306 这类为无源 MOS 架构芯片，低电平是通过 MOS 导通传输到后级，不具备主动低电平驱动能力，同时不隔离总线前后级电容。更适合应用于板内通讯轻负载场景，低功耗应用等场景。

b) TPT29617A 级联注意事项

TPT29617A 的 2 侧低电平输出 V_{OL} 存在静态偏移 ($V_{OL}=0.5V$) 且高于自身 V_{IL} ，因此级联时不允许将两个芯片的 2 侧接在一起，否则 0.5V 低电平输出会被第二级 TPT29617A 的 2 侧识别为高电平，导致通讯失败。只允许将 1 侧和 2 侧连接在一起或将两个 1 侧连接在一起。

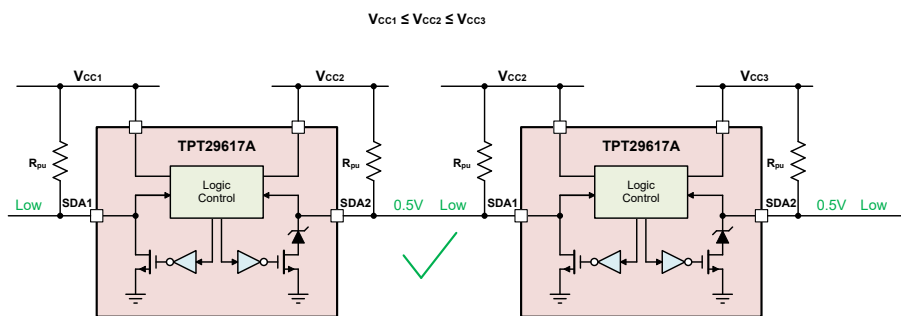


图 8-27 TPT29617A 正确级联接法 1

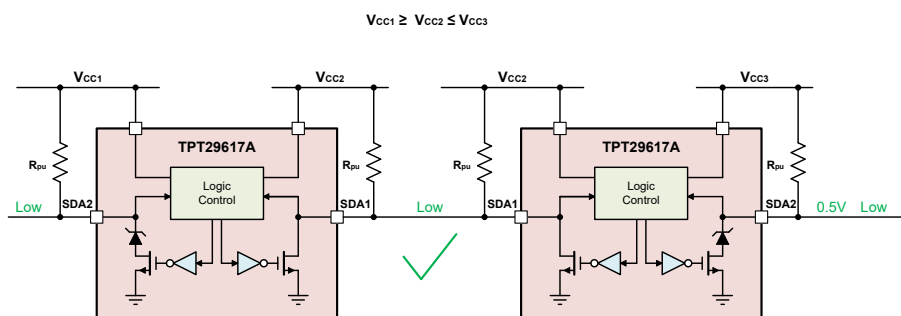


图 8-28 TPT29617A 正确级联接法 2

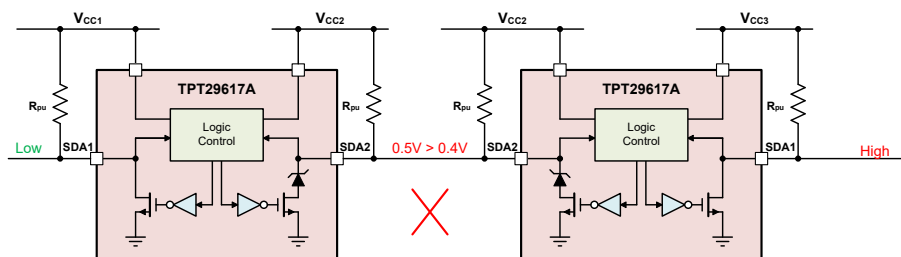


图 8-29 TPT29617A 错误级联接法

8.7 热插拔缓冲器

8.7.1 概述

I²C 缓冲器是一类具备信号重驱动和电容隔离能力的器件，将总线分割为两侧独立区段，各自可驱动更高的容性负载。在此基础上，部分缓冲器进一步集成了热插拔功能，专门解决板卡在系统不断电情况下插入或拔出时引发的总线可靠性问题。

热插拔是指外部板卡从带电运行的背板中插入或拔出，而无需关闭系统电源。此功能在服务器、通信基站、存储阵列等不允许或难以断电维护的高可靠性系统中尤为重要。热插入瞬间，带电背板向未上电板卡的寄生电容注入浪涌电流，若此时背板总线为高电平，该浪涌可能在总线上产生虚假的下降沿，使总线上的目标器件误收到错误的时钟或数据脉冲，造成通讯错误，甚至导致总线阻塞挂死。

传统分立式热插拔方案通常采用 I²C 开关配合插入检测信号，由处理器检测到板卡连接后再使能对应通道接入总线，此方式可防止毛刺干扰背板，但需软件参与。而在板卡侧集成专用热插拔缓冲器（如 TPT29511），可以硬件自动管理连接时序，广泛用于主要应用于服务器、电信交换设备、基站及工业自动化等需要频繁带电插拔 I/O 卡或模块的场景，能有效提升系统的可靠性和维护便利性。

8.7.2 工作机制

TPT29511 是一款专为热插拔板卡设计的双通道双向 I²C 缓冲器。器件将总线分为 IN 侧（面向背板，连接系统总线）和 OUT 侧（面向板卡，连接本地 I²C 器件），两侧通过内部有源驱动电路实现电容隔离与信号重驱动。在插入过程中，IN 侧和 OUT 侧连接断开，而不会在板卡上发生总线争用。在插入过程中，会对 SDA 和 SCL 线路预充电至 1V，从而尽可能减小对器件的寄生电容充电所需的浪涌电流。插入完成且检测到背板上出现总线空闲情况或者停止命令后，通道连接。

a) 电平传输

TPT29511 的高电平通过电阻上拉维持。低电平则为重新驱动，其具备特殊的方向识别机制来防止总线锁死，输出侧驱动低电平相比输入侧存在 100mV 的偏移抬升。例如，输入侧器件低电平为 0.1 V，经过 TPT29511 重新驱动后，输出约为 0.2 V。

b) 1V 预充电电路

TPT29511 在上电完成后且 SDA 和 SCL 脚上拉到 I²C 总线电源之前，这两路引脚会被预充电到 1V。预充电电路为弱驱动，具备 100k Ω 内阻。该功能有助于最大程度地限制浪涌电流，减小热插拔事件带来的总线毛刺干扰。预充电功能在器件总线连接且上拉完成后自动关闭。

c) 通道连接

TPT29511 的 IN 侧与 OUT 侧之间建立连接前须同时满足三个条件：（1）VCC 已上电且内部电路完成初始化；（2）OUT 侧 SDA 和 SCL 均为高电平（确保板卡上器件已就绪且未将总线拉死）；（3）在 IN 侧的 SDA 或 SCL 上检测到停止条件或总线空闲状态（确保背板总线当前不处于活动传输中，避免板卡器件误收到不完整帧）。三个条件全部满足后，IN 侧与 OUT 侧连通，板卡接入背板总线。READY 引脚为开漏输出，连通后释放为高电平（需外部上拉），可供处理器查询板卡就绪状态。

d) 上升时间加速器

上升时间加速器

TPT29511 在两侧均集成上升时间加速器（RTA）。当检测到信号上升沿压摆率超过设定阈值（0.6 V/us）且电压幅度超过 0.8V 时，RTA 短暂输出额外上拉电流，迅速拉升总线。如下图所示：

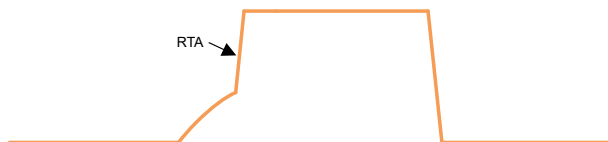


图 8-30 TPT29511 上升时间加速波形

8.7.3 热插拔应用设计

TPT29511 背板热插拔应用以及金手指设计如下所示

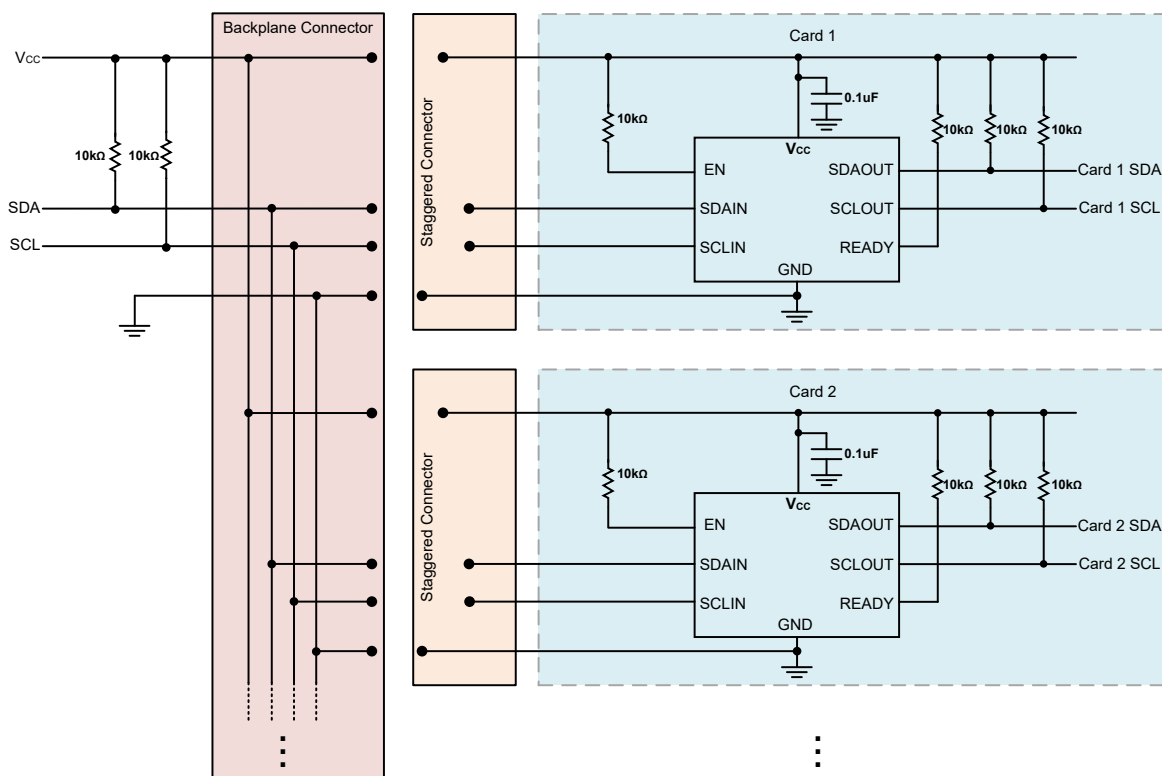


图 8-31 TPT29511 热插拔应用

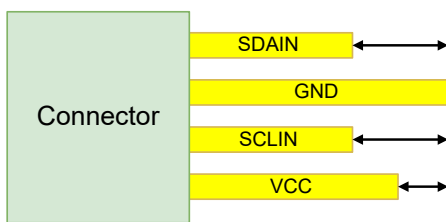


图 8-32 板卡金手指设计

热插拔设计需要注意以下事项：

- 设计板卡金手指长度时，需要保证 GND、VCC、SDAIN/SCLIN 的先后连接顺序。如果 SDAIN 或 SCLIN 线路与 VCC 和 GND 同时连接至背板，那么，SDA/SCL 来不及被预充电到 1V，连接时背板 I²C 总线可能会被瞬间浪涌电流下拉出低电平毛刺，干扰背板通讯。
- IN 侧的上拉电阻需要加在背板上，不能放在板卡侧。若将其置于板卡侧，当 VCC 上电后 SCLIN 和 SDAIN 会同步被电阻拉高，TPT29511 检测到 IN 侧和 OUT 侧均处于空闲状态后，将关闭预充电电路并开启通道连接。这样一来，通道在板卡完全插入之前就已建立，无法满足“插入完成后再连接总线”的设计需求。
- 器件 IN 侧面向插拔侧，IN 侧和 OU 侧不建议互换。TPT29511 仅在器件的 IN 侧进行总线空闲/停止条件检测，而“OUT 侧”仅检测总线空闲。因此如果互换使用，OUT 侧插入背板无法通过检测 STOP 信号完成快速连接。而 IN 侧可以通过检测 STOP 信号来快速连通总线。
- TPT29511 不建议多级级联，由于其每级都会带来 100mV 的低电平偏移，多级级联后低电平被抬高，当抬升至 0.7V 附近时，容易造成误触发上升沿加速。因此一般情况下，仅需将 TPT29511 放置在板卡侧，背板侧并不需要，单级 TPT29511 已经可以满足应用需求。

当满足上述要求时，TPT29511 热插入过程如下：

1. 板卡插入插槽，GND 触点率先接通，随后 VCC 触点接通，板卡侧上电。
2. TPT29511 上电后，1V 预充电电路启动，将 SDAIN 和 SCLIN 充电至 1V。同时 IN 侧和 OUT 侧不满足连接条件，通道保持关断。
3. SDA/SCL 触点最后接通背板总线。由于预充电已完成，因此插入时 IN 侧连接瞬间的浪涌电流被抑制，可以防止总线出现低电平毛刺干扰，影响背板通讯。
4. SDAIN/SCLIN 连接后，将被背板的上拉电阻拉高到空闲状态，或者当背板侧发出 STOP 信号后，IN 侧与 OUT 侧通道开启连接。板卡接入背板成功，处理器可正常访问板卡上器件。

8.7.4 背板分立热插拔方案

除了使用 TPT29511 这类热插拔缓冲器外，还可以使用 I²C 开关配合 IO 检测实现分立热插拔方案，如图 8-33 所示。

例如，I²C 开关 TPT29548L 配合 IO 拓展芯片 TPT29555A 实现热插入功能。其中，TPT29548L 放置在背板边缘，用来连接外部板卡。在板卡插入前，其相应的通道保持关闭。因此插拔时产生的毛刺干扰，不会影响到另一侧总线的正常通讯。TPT29555A 用来做插入检测，其 IO 端口有 100k Ω 的内部上拉电阻，板卡插入前 IO 为高电平。插拔完成后，IO 被外部板卡拉低，因此将触发 TPT29555A 的 $\overline{\text{INT}}$ 中断输出。背板主控检测到中断后，即可读取 TPT29555A 所有端口输入电平状态，确定是某个通道板卡插入后，再控制 TPT29548L 开启相应的通道，实现和板卡通信。通过上述流程，可以实现热插拔功能。TPT29548L 有 8 个通道，可以支持 8 块板卡连接。

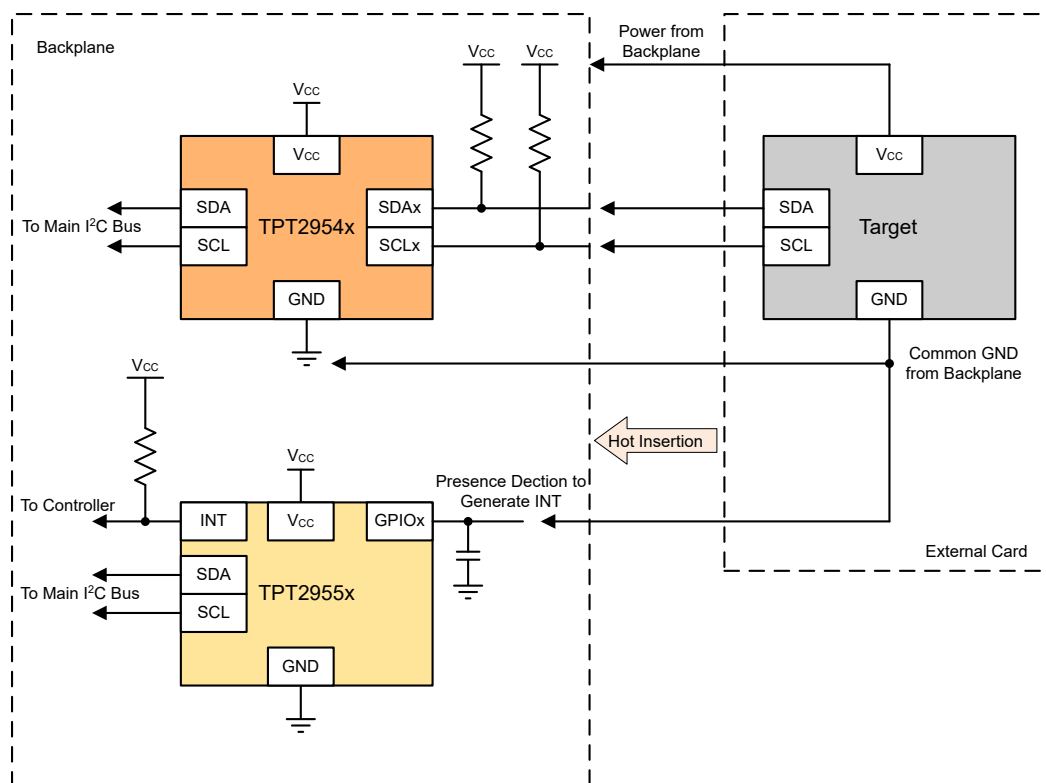


图 8-33 背板分立热插拔方案

TPT29548L 可以替换为其他 I²C 开关，或者带使能引脚的 I²C 中继器件。TPT29555A 也可以替换其他带中断的 GPIO 扩展芯片，或者直接使用背板主控的 GPIO。

此外，也可直接使用 TPT29545 这类自带中断功能的 I²C 开关实现背板热插拔方案，无需额外的 IO 拓展芯片，如图 8-34 所示。其具有 1:4 的 I²C 总线拓展通道和 4:1 的 $\overline{\text{INT}}$ 上传通道。其中 $\overline{\text{INTx}}$ 通道可以用来做插入检测，当板卡连接后将下游 $\overline{\text{INTx}}$ 拉低，将触发上游 $\overline{\text{INT}}$ 输出，背板主控检测到中断后，可以读取 TPT29545 寄存器获取中断输入状态，并开启对应的 I²C 通道。

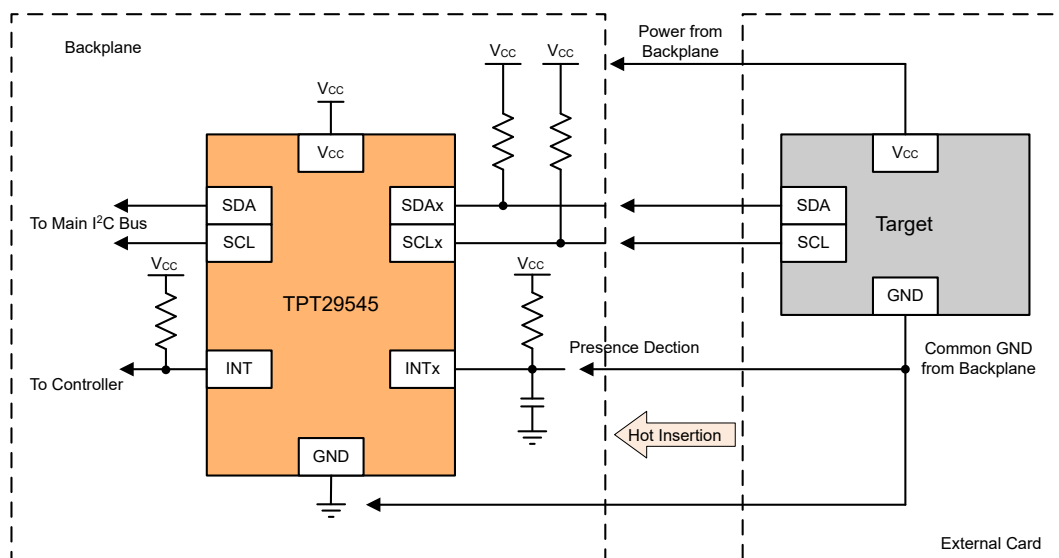


图 8-34 使用 TPT29545 的背板分立热插拔方案

8.7.5 应用注意事项

a) TPT29511 级联

TPT29511 不建议多级级联。TPT29511 低电平输出存在 100mV 的偏移，即每经过一级，低电平就会抬升 0.1V。多级级联可能导致低电平抬升超过后级器件 V_{IL} 阈值。另外，如果电平抬升至 TPT29511 上拉加速阈值电压附近，还可能引起上拉加速误开启，产生异常数据和时钟。由于 TPT29511 本身支持上升沿加速功能，应用中一般使用一级即可驱动较重容性负载，最多的情况下建议级联不要超过 2 级。

TPT29511 与其他器件配合使用时需要注意，TPT29617A 的 2 侧不建议与 TPT29511 级联。TPT29617A 的 2 侧输出低电平 0.5V，经过 TPT29511 后输出抬升至 0.6V，离上拉加速阈值电压较近。如果总线还存在串阻或其他开关器件，低电平抬升更高，可能导致无法满足后级器件的 V_{IL} 要求。使用时更推荐将 TPT29511 放置在 TPT29617A 的 1 侧，或者和 TPT29306 这类不存在低电平静态偏移的电平转换器件搭配使用。

9. 修订记录

日期	版本	注释
2026.07.15	Rev.A.0	初版发布

10. 参考资料

1. NXP, *I²C-bus specification and user manual*
2. 3PEAK, *TPT29548L 1.8-V 8-channel I²C Switch with Reset Datasheet*
3. 3PEAK, *TPT29555A I²C to 16-Bit GPIO Expander with Interrupt Datasheet*
4. 3PEAK, *TPT29535A I²C to 16-Bit GPIO Expander with Interrupt Datasheet*
5. 3PEAK, *TPT29554A I²C to 8-Bit GPIO Expander with Interrupt Datasheet*
6. 3PEAK, *TPT29508 I²C to 8-bit GPIO Expander with Voltage-level Translation Datasheet*
7. 3PEAK, *TPT29306 Dual Bidirectional I²C Bus and SMBus Voltage-Level Translator Datasheet*
8. 3PEAK, *TPXF0102 2-Bits Bidirectional Multi-Voltage Level Translator for Open-Drain and Push-Pull Application Datasheet*
9. 3PEAK, *TPXF0204 4-Bits Bidirectional Multi-Voltage Level Translator for Open-Drain and Push-Pull Application Datasheet*
10. 3PEAK, *TPT29606 Dual Bidirectional I³C/I²C-bus and SPI Voltage-level Translator Datasheet*
11. 3PEAK, *TPT20102 2-bit Bidirectional Level Shifter for Open-Drain and Push-Pull Datasheet*
12. 3PEAK, *TPT29617A Dual Channel Bidirectional I²C Bus Level Shift and Repeater Datasheet*
13. 3PEAK, *TPT29511H Hot-Swap I²C-BUS Buffer Datasheet*

11. 声明与提示

著作权© 思瑞浦 2012-2026，版权所有。

商标。本文档/材料所包含的任何思瑞浦和 3PEAK 的商号、商标、图形标志和域名，均为思瑞浦所有。未经思瑞浦事先书面许可，不得以任何形式将其复制、修改、出版、传输或发布。

性能信息。本文档/材料中所包含的产品性能测试指标和额定值测量为特定环境下的设计仿真值或实际测试值。任何测试环境或仿真环境的不同，包括但不限于测试方法、测试流程、测试温度等的不同，都可能影响产品的实际性能。

免责声明。思瑞浦“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出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。除非另有书面说明，否则，思瑞浦提供的产品并非设计用于任何危及生命的场景，包括关键医疗应用、汽车安全关键系统、航空、航天或任何故障可能导致人身伤害、生命丧失或重大财产损失的情形。思瑞浦不对任何此类未经授权的使用承担任何责任。