

引言

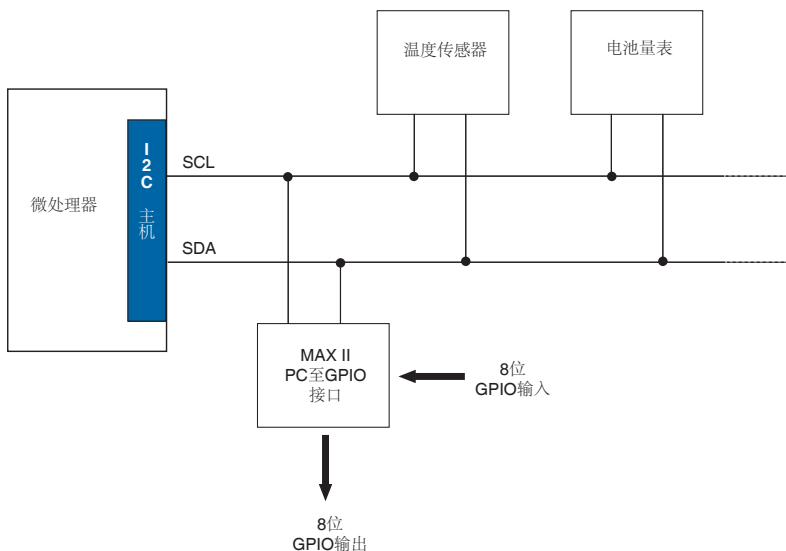
本设计实例展示了 Altera® MAX® II CPLD 通过工业标准 I²C 总线提供通用 I/O (GPIO) 引脚扩展的能力。为了减小封装尺寸和引脚数量, 很多微处理器系统限制了通用 I/O 的数量。然而, 如果系统有 I²C 接口, 那么本设计可以通过 I²C 总线来加入更多的 GPIO 引脚。借助于 MAX II CPLD, 增加的这些 GPIO 引脚要比微处理器的 I/O 引脚功耗低。

GPIO 引脚扩展和 I²C

在某些情况下, 可能需要通过较长的 PCB 走线来访问系统中的 GPIO 引脚 (例如蜂窝电话中不同部分的连接)。I²C 接口是 2 线系统, 本设计通过普通的 2 线走线, 在远端提供多个输入和输出引脚。这提高了设计灵活性以及整个系统的物理紧凑性, 还减小了封装尺寸和引脚数量。

可以通过这些通用输出引脚来连接并控制风扇控制器、LED 状态显示和状态指示器等设备。同样的, 复位引脚和按钮开关等设备可以直接连接到 CPLD 的通用输入上, 实现各种应用。

图 1. 通过 I²C 总线实现 GPIO 引脚扩展



GPIO 引脚扩展的 I²C 接口

MAX II CPLD 用作 I²C 总线从机，在其 I²C 接口上有两个引脚：I²C 时钟 SCL 和 I²C 数据线 SDA。作为 I²C 主机的主系统和 MAX II (用作 I²C 从机) 进行通信。CPLD 为主系统提供 8 个通用输入端口和 8 个通用输出端口。GPIO 引脚并行接收 I²C 总线串行发送的数据。这样，可以同时读写所有 8 个通用 I/O。

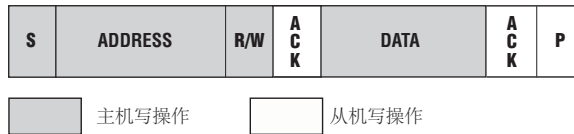
I²C 接口

对于 I²C 接口，CPLD (I²C 从机) 提供一个符合常用 I²C 协议的内置 7 位地址。主机发送起始信号，然后是 7 位地址和一个 R/W 位。当 I²C 总线上广播的地址和从机设备的地址匹配时，根据主机发送的读或者写信号，设备发送一个 ACK (应答) 信号，然后是数据。随后是另一个 ACK 信号。以此不断进行数据交换，直到主机发送 Stop (P) 信号。

表 1. I²C 接口引脚说明

信号	目的	方向
SCL	I ² C 时钟	输出
SDA	I ² C 串行数据	双向

图 2. I²C 信号格式



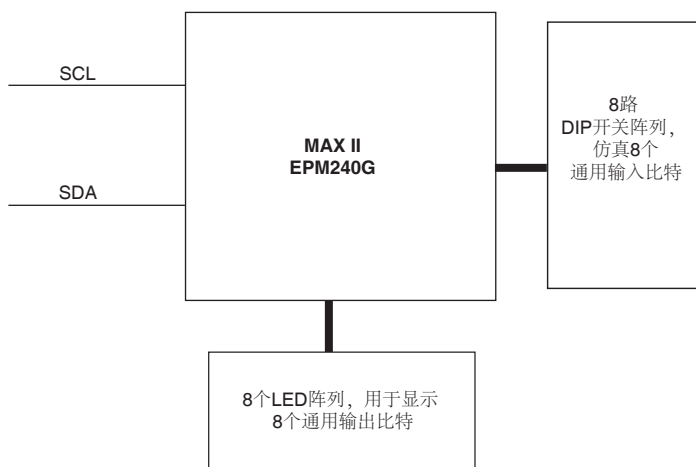
S = Start (SCLK 高电平, SDA 高电平至低电平转换)
 R/W = Read/Write (1 为读操作, 0 为写操作)
 ACK = Acknowledgement (SDA 被接收器置为低电平)
 P = Stop (SCLK 高电平, SDA 低电平至高电平转换)
 默认从机地址 = 0000000 (00h)

GPIO 接口

当主机发出写命令 ($R/W = 0$) 时, I²C 总线接收到的数据被用于更新通用输出引脚, 直到接收到停止或者重新启动命令。同样的, 当 I²C 主机发出读命令 ($R/W = 1$) 时, 在 ACK 位对通用输入引脚上的数据进行采样, 并通过 I²C 总线串行发送。一直持续这一过程, 直到主机发出停止或者重新启动命令。

信号	目的	方向
8 位输入	通用	输入
8 位输出	通用	输出

图 3. GPIO 引脚扩展演示电路



设计实现

该设计实例可以采用 EPM240G 或者其他 MAX II CPLD 来实现。实现过程涉及到使用设计实例源代码, 为 MAX II CPLD GPIO 分配 I²C 总线和 GPIO 引脚扩展输入和输出。在 MDN-B2 演示板上采用 I²C 仿真器演示了 GPIO 引脚扩展。使用 PC 并口来生成该仿真器, 通过接口硬件来生成 I²C 兼容二线总线。Dallas 半导体公司的 Maxim 应用笔记 AN3230 详细说明了怎样建立 I²C 环境, 可以从下面的链接下载该应用笔记:

www.maxim-ic.com/appnotes.cfm/an_pk/3230。

除了能够从这一 URL 下载免费的软件外，还可以从下面的链接下载并口 I²C 设置的相应免费软件程序：

http://files.dalsemi.com/system_extension/AppNotes/AN3315/ParD_S2W.exe

这一实用程序采用了并口及其硬件和 MAX II CPLD 进行接口，并提供 I²C 2 线系统需要的 SDA 和 SCL 连接。完成后，该设计使 MDN-B2 演示板 (通过 DIP 开关设置) 的输入能够到达 I²C 主机。同样，I²C 主机发送的数据出现在 MAX II CPLD 的 GPIO 输出端口 (连接到演示板的 LED) 上。对于该演示，I²C 主机是运行并口 I²C 软件 PC 的用户接口。

下面详细介绍了怎样在 MDN-B2 演示板上实现本设计实例。表 3 列出了本设计实例的 EPM240G 引脚分配。

表 3. MDN-B2 演示板设计实例			
EPM240G 引脚分配			
信号	引脚	信号	引脚
SCLK	引脚 39	SDA	引脚 40
GPIO_output[0]	引脚 69	GPIO_output[1]	引脚 70
GPIO_output[2]	引脚 71	GPIO_output[3]	引脚 72
GPIO_output[4]	引脚 73	GPIO_output[5]	引脚 74
GPIO_output[6]	引脚 75	GPIO_output[7]	引脚 76
GPIO_input[0]	引脚 55	GPIO_input[1]	引脚 56
GPIO_input[2]	引脚 57	GPIO_input[3]	引脚 58
GPIO_input[4]	引脚 61	GPIO_input[5]	引脚 66
GPIO_input[6]	引脚 67	GPIO_input[7]	引脚 68

在 Quartus® II 软件中，把未使用的引脚分配为 **input-tristated**。您还必须使能 SCLK 和 SDA 引脚上的 **Auto Open-Drain** 设置。对此，在 Assignments 菜单中，单击 **Settings**，然后选择 **Analysis and Synthesis Settings**，使能 **Auto Open-Drain** 设置。设置完成后是进行编译。

设计说明

按照以下步骤在 MDN-B2 演示板上演示本设计：

1. 使用滑动开关 SW1 打开演示板电源。
2. 通过演示板上 JTAG 插头 JP5 和普通编程电缆 (ByteBlasterTMII 或者 USB-BlasterTM), 把设计下载到 MAX II CPLD 中。
3. 在编程启动前和启动过程中, 保持演示板上 SW4 的按下状态不变。编程完成后, 关断电源, 拔下 JTAG 连接器。
4. 按照以下步骤在 PC 上建立并口驱动 I²C 环境:
 - a. 下载 Maxim 并口实用程序等软件工具实现和 I²C 协议的从机进行通信。安装并口软件。(这一实例使用了 **ParDS2W.exe** 程序)
 - b. 对于这一并口程序, 您必须安装并口驱动程序来使能对 Windows XP 或者 Windows 2000 并口的访问。您可以从 Direct-IO 下载该驱动程序: www.direct-io.com/Direct-IO/directio.exe
 - c. 安装完成后, 必须配置 Direct-IO 程序。打开 Windows 控制面板, 单击 Direct IO 图标。输入并口的起始和结束地址, 这通常是 378 至 37F; 但是要查看下面的设置来确定 PC 的并口地址: Control Panel/System/Hardware/Device Manager/Ports/ECP Printer port (LPT)/Resources。
 - d. 启动 PC 时, 在 BIOS 设置中把并口配置为 ECP。
 - e. 然后, 在 Direct IO 控制面板中选择 **Security** 标签, 浏览 **ParDS2W.exe** 程序路径。单击 **Open**, 然后单击 **Add**, 加入程序。在 **Allowed Processes** 域中出现该程序的路径。单击 **OK**。
 - f. 安装 MDN-B2 演示板提供的并口 I²C 软件狗。如果需要, 使用延长线延长并口连接, 以靠近演示板。
 - g. 连接 I²C 并口软件狗的 4 针插座和演示板的 I²C 插头 (JP3), 插座的红色标记对上 JP3 插头的引脚 1。
 - h. 打开 ParDS2W 程序, 选择 PC 合适的并口地址 (配置 Direct IO 时可以看到), 把 **2-Wire Device Address** 设置为 00h。
 - i. 最后, 在 **Test Circuit** 标签下测试 I²C 设置, 观察在 Status 窗口中是否出现 Test PASS 消息。如果出现, 则表明现在建立了 I²C 环境。
5. 通过并口实用程序, 利用 2 线功能, 进行 I²C 读写操作。

6. 进行 I²C 写操作时，单击 **Start**，然后单击 **Write Byte**。在靠近 **Write Data** 的区域中键入一个十六进制字节，单击 **Write Data**。观察 8 个红色 LED 上相应的数值。每次写操作后，单击 **Stop**。
7. 同样的，单击 **Start**，然后单击 **Read Byte**，进行读操作。Read 窗口显示演示板上 SW5 开关的设置。每次读操作后，单击 **Stop**。

源代码

本应用笔记所介绍的设计实例采用了 Verilog HDL 来实现，成功地运行在 MDN-B2 演示板上。下面的链接提供源代码、测试台文件以及完整的 Quartus II 工程：

www.altera.com/literature/an/an494.zip

结论

正如本设计实例所示，MAX II CPLD 是实现 I²C 等工业标准接口非常好的选择。其低功耗以及易于上电特性及其内部振荡器功能使其成为需要 I²C 接口来提供 GPIO 引脚扩展等应用理想的可编程逻辑器件选择。

其他资源

- MAX II CPLD 主页：
www.altera.com/products/devices/cpld/max2/mx2-index.jsp
- MAX II 器件资料：
www.altera.com/literature/lit-max2.jsp
- MAX II 关断设计：
www.altera.com/support/examples/max/exm-power-down.html
- MAX II 应用笔记：
[AN 422: 利用 MAX II CPLD 实现便携式系统的功耗管理](#)
[AN 428: MAX II CPLD 设计指南](#)

文档版本历史

表 4 列出了本应用笔记的版本历史。

表 4. 文档版本历史		
日期和文档版本	进行的改动	对改动的总结
2007 年 12 月, 1.0 版	初次发布	—



101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support:
www.altera.com/support/
Literature Services:
literature@altera.com

版权 © 2007 Altera 公司。保留所有版权。Altera、可编程解决方案公司、程式化 Altera 标识、专用器件名称和其他所有其他专有商标或者服务标记，除非特别声明，均为 Altera 公司在美国和其他国家的商标和服务标记。所有其他产品或者服务名称的所有权属于其各自持有人。Altera 产品受美国和其他国家多种专利、未决应用、模板著作权和版权的保护。Altera 保证当前规范下的半导体产品性能与 Altera 标准质保一致，但是保留对产品和服务在没有事先通知时的升级变更权利。除非与 Altera 公司的书面条款完全一致，否则 Altera 不承担由此处所述信息、产品或者服务导致的责任。Altera 建议客户在决定购买产品或者服务，以及确信任何公开信息之前，阅读 Altera 最新版的器件规范说明。

