

引言

本应用笔记介绍 Altera® MAX® II CPLD 通过工业标准系统管理总线 (SMBus) 来提供通用 I/O 引脚扩展功能。

通过 SMBus 实现 GPIO 引脚扩展

为了缩小封装尺寸，减少引脚数量，很多微处理器系统限制了通用 I/O (GPIO) 引脚的数量，但是如果系统有 SMBus 接口，那么本设计可以通过 SMBus 来提供更多的 GPIO 引脚。而且，使用 MAX II CPLD 时，这些增加的 GPIO 引脚要比微处理器的 I/O 引脚功耗低。

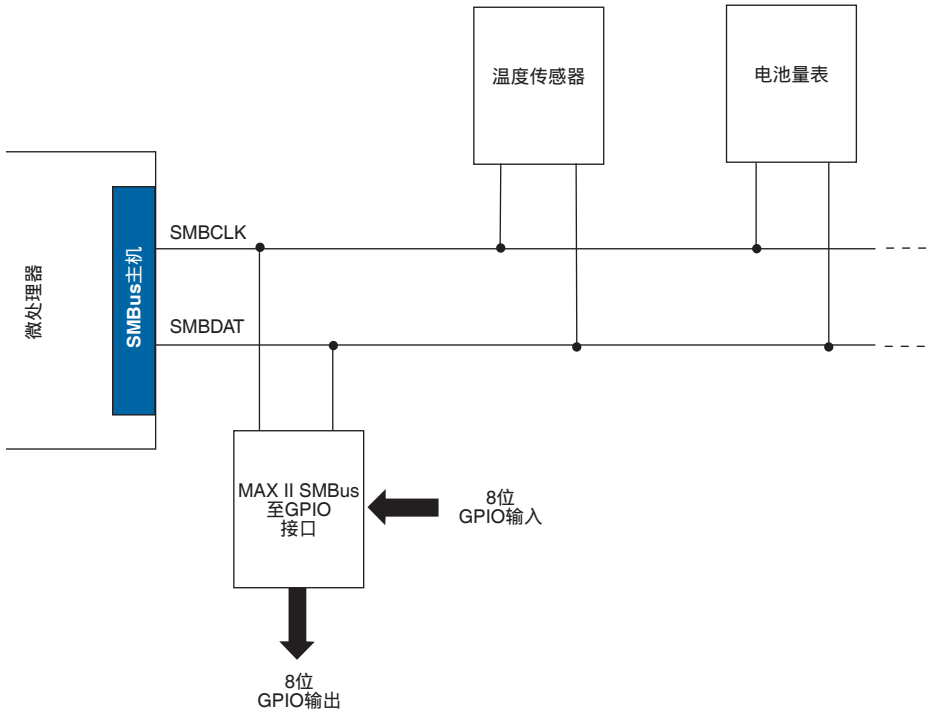
在某些情况下，您可能需要远程访问嵌入式系统中的 GPIO 引脚。由于 SMBus 接口是 2 线系统，本设计在远端提供多个输入和输出引脚，不需要其他的连线。这提高了设计灵活性，降低了复杂度，减少了软件开销。

您可以通过这些通用输出引脚来连接并控制外设控制器、LED 和其他状态指示器。同样的，您还可以在前面板上连接用户控制、按键以及开关和 CPLD 提供的通用输入。

基于 MAX II CPLD 的 SMBus 实现 GPIO 引脚扩展

MAX II CPLD 用作 SMBus 从机，在其 SMBus 接口上有两个引脚：SMBus 时钟 SMBCLK 和数据线 SMBDAT。作为 SMBus 主机的主系统和 MAX II (用作 SMBus 从机) 进行通信。CPLD 为主机提供 8 个通用输入端口和 8 个通用输出端口。GPIO 引脚并行接收由 SMBus 串行发送的数据。这样，可以同时读写所有 8 个通用 I/O (参见图 1)。

图 1. 通过 SMBus 进行 GPIO 引脚扩展



SMBus 接口

对于 SMBus 接口，CPLD 提供一个符合常用 SMBus 协议的内置 7 位地址。主机发送起始信号，然后是 7 位地址和一个读写位。当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 DATA。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 对 SMBus 接口进行了说明。

表 1. SMBus 接口说明

信号	目的	方向
SMBCLK	时钟	输入 (从机)
SMBDAT	串行数据	双向

图 2 所示为 SMBus 信号格式。

图 2. SMBus 信号格式

S	ADDRESS	R/W	ACK	DATA	ACK	P
---	---------	-----	-----	------	-----	---

S = Start (SMBCLK高电平, SMBDAT高电平至低电平转换)

R/W = Read/Write (1为读操作, 0为写操作)

ACK = Acknowledge (SMBDAT被接收机置为低电平)

P = Stop (SMBCLK高电平, SMBDAT低电平至高电平转换)

默认从机地址 = 0000000

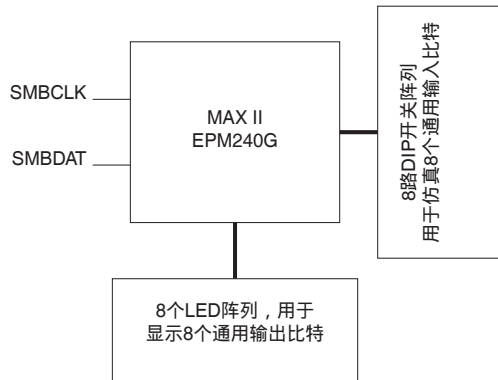
GPIO 接口

当主机发出写命令 (R/W = 0) 时, SMBus 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, 当总线主机发出读命令 (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 SMBus 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。表 2 对 GPIO 引脚进行了说明。图 3 所示为 GPIO 引脚扩展演示电路。

表 2. GPIO 引脚说明

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 EPM240 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 SMBus 总线和 GPIO 引脚扩展输入和输出。在 MDN-B2 演示板上采用 2 线仿真器演示了 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 SMBus 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这一环境, 可以从下面的链接中下载该应用笔记:

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

还可以从下面的链接下载相应的并口至 2 线接口设置软件:

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 SMBus 2 线系统需要的 SMBDAT 和 SMBCLK 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 SMBus 主机。同样, SMBus 主机发送的数据出现在 MAX II CPLD 的 GPIO 输出端口(和演示板的 LED 连接)上。对于该演示, SMBus 主机是运行并口至 2 线软件的 PC 的用户接口。

表 3 列出了怎样在 MDN-B2 演示板上实现该设计实例。

信号	引脚
SMBCLK	引脚 39
GPIO_output[0]	引脚 69
GPIO_output[2]	引脚 71
GPIO_output[4]	引脚 73
GPIO_output[6]	引脚 75
GPIO_input[0]	引脚 55
GPIO_input[2]	引脚 57
GPIO_input[4]	引脚 61
GPIO_input[6]	引脚 67
SMBDAT: pin 40	引脚 40
GPIO_output[1]	引脚 70
GPIO_output[3]	引脚 72
GPIO_output[5]	引脚 74
GPIO_output[7]	引脚 76
GPIO_input[1]	引脚 56
GPIO_input[3]	引脚 58
GPIO_input[5]	引脚 66
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 SMBDAT 引脚的 **Auto Open-Drain** 设置。在 Assignments 菜单中, 选择 **Settings**。出现 **Settings** 对话框。在 **Category** 列表中, 选择 **Analysis & Synthesis Settings**。打开 **Auto Open-Drain Pins**。在 Quartus II 软件中分配引脚并进行设置后, 对设计进行编译。

演示说明 (在 MDN-B2 演示板上演示该设计):

- 打开演示板电源 (使用滑动开关 SW1)。
- 通过演示板上 JTAG 插头 JP5 和普通编程电缆 (ByteBlaster II 或者 USB-Blaster), 把设计下载到 MAX II CPLD 中。
- 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。一旦完成, 关断电源, 去掉 JTAG 连接器。
- 在 PC 上建立并口驱动 SMBus 环境:

- 下载 Maxim 并口实用程序等软件工具实现和 SMBus 协议的从机进行通信。安装并口软件。这里介绍的实例使用了程序 **ParDS2W.exe**:
http://files.dalsemi.com/system_extension/AppNotes/AN3315/ParDS2W.exe
- 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。Direct-IO (www.direct-io.com) 提供您可以使用的典型驱动程序, 从下面的链接下载:
www.direct-io.com/Direct-IO/directio.exe
- 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 (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。
- 如果您的并口被配置为 ECP 以外的其他类型, 启动 PC 时, 在 BIOS 设置中把并口配置为 ECP。
- 在 Direct IO 控制面板中选择 **Security Tab**, 浏览 **ParDS2W.exe** 程序所在的路径。单击 **Open**, 然后单击 **Add**。在 **Allowed Processes** 域中出现该程序的路径。单击 **OK**, 关闭控制面板窗口。
- 安装 MDN-B2 演示板提供的并口 I2C/SMBus 软件狗。如果需要, 使用延长线延长并口连接, 以靠近演示板。
- 连接 I2C/SMBus 并口软件狗的 4 针插座和演示板的插头 (JP3), 插座的红色标记对上 JP3 插头的第 1 引脚。
- 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。
- 最后, 利用 **Test Circuit** 标签来测试设置, 观察在 **Status** 窗口中是否出现 **Test PASS** 消息。如果出现, 则表明现在建立了需要的 2 线环境。
- 利用并口实用程序 **2-Wire Utility** 部分的 **One byte Write/Read** 一节, 对 GPIO 输入 (DIP 开关 SW5) 进行读操作, 还可以进行写操作, 对演示板的 8 个 LED 阵列写入要显示的 8 位数据。
- 从 DIP 开关 SW5 读取数据时, 按下这部分的一个 **Read** 按钮即可 (由于该应用程序是从存储器读取数据, 因此, 在进行 **Read** 之前, 必须在 **Address** 域中输入任意 2 位十六进制数值)。靠近它的数据区将根据 DIP 开关设置显示 8 位位置。
- 在进行写操作时, 在并口应用程序这一部分的 **Address** 域输入任意有效的十六进制数据 (**Data** 域必须填充任意 2 位十六进制数值, 但是在前后关系上并不重要), 然后按下 **Write** 按钮。和数据对应的 LED 被点亮。

- 将要写入的数据输入到 **Address** 域，然后执行 **Read** 按钮操作，可同时进行读写操作。程序把 **Address** 域数据发送到 SMBus 从机设备，然后立即执行读操作。之所以能够这样，是因为应用程序能够读取存储器从机设备，但是在读取之前，必须先写入从机设备的地址。
- **2-Wire Function** 部分还能够进行读写操作。但是，不能使用该功能。**2-Wire Function** 手动驱动 SMBus 操作（启动 / 停止 / 读 / 写等）。因为是以手动速度来输入并执行每一功能，因此手动操作较慢。较慢的速度使 SMBCLK 线的低电平时间大于 25 ms，导致 SMBus 从机自己复位。为避免这一点，该演示中应使用 **2-wire Utility** 部分。

源代码

本文档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/an484_design_example.zip

结论

正如本设计实例所示，MAX II CPLD 是实现 SMBus 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 SMBus 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

下面列出了其他资源：

- 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 428：MAX II CPLD 设计指南](#)
[AN 422：利用 MAX II CPLD 实现便携式系统的功耗管理](#)

版本历史

表 4 列出了本应用笔记的版本历史。

日期和版本	进行的改动	注释
2007 年 12 月, 1.0 版	初次发布	—



101 Innovation Drive
San Jose, CA 95134
www.altera.com
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、专用器件名称和所有其他专有商标或服务标记, 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、产品或服务导致的责任。Altera 建议客户在决定购买产品或服务, 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。



I.S. EN ISO 9001