

引言

本应用笔记讨论非易失信息的存储问题。大部分 CPLD 使用串行 EEPROM 来存储非易失信息, 而 MAX[®] II CPLD 是唯一具有用户闪存 (UFM) 的 CPLD, 它支持用户存储高达 8 Kbits 的非易失信息。除了可编程以外, UFM 还支持串口和并口以及其他专用协议。本应用笔记解释了怎样在 UFM 中高效地存储和读取信息, 使用 I²C 协议实现 MAX II CPLD 的 UFM 接口和访问。

用户闪存

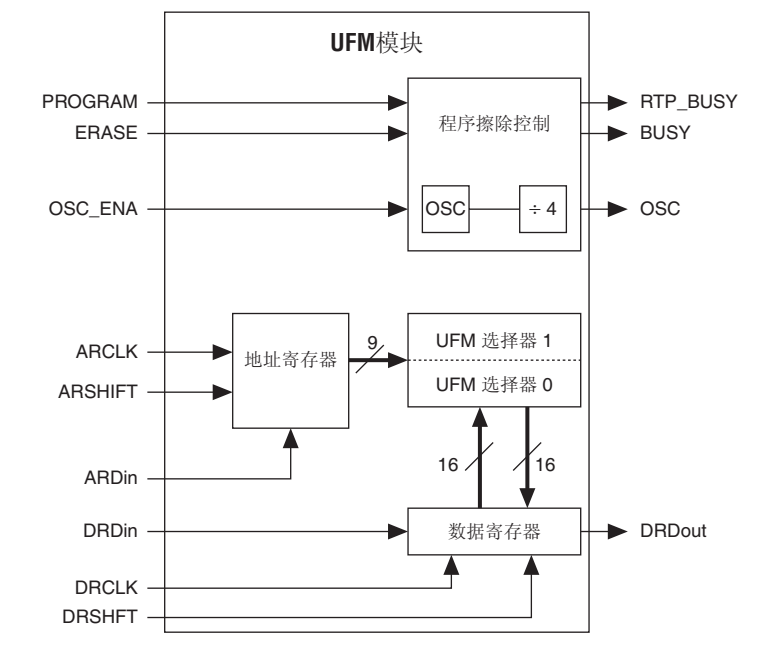
以下列出了 UFM 的实际应用和特性:

- 您可以利用 UFM 来存储关键的非易失信息, 例如 ASSP 或者处理器配置比特、电路板生产过程中的电子 ID 信息, 还可以存储处理器上电时在 LCD 上显示的信息。
- 其特性包括并口和串口, 例如串行通信接口 (SCI)、串行外设接口 (SPI)、内置集成电路 (I²C)、Microwire 等其他专用协议。MAX II 器件在接口上比商用 EEPROM 器件更加灵活。
- UFM 的内部振荡器能够满足所有设计的时钟需求; 它不需要外部时钟电路, 因此节省了空间和成本。

在设计电路板时, 您可以采用 MAX II CPLD 来合并逻辑和存储器, 从而缩短了芯片至芯片延时, 减小了电路板面积, 降低了系统总成本。

UFM 划分成两个区, 每个 4 Kbits。地址寄存器指示数据写入和读出的 UFM 存储器地址。数据寄存器保存向 UFM 写入或者读出的数据。程序擦除控制模块用于对 UFM 的编程或者擦除, 还可以使能内部振荡器。图 1 所示为 UFM 的结构图。

图 1. 用户闪存 (UFM) 结构图



使用 MAX II 器件的 UFM

UFM 被分成两个区：0 区和 1 区。每一区都是 4096 比特，地址范围分别为 000h 到 0FFh 和 100h 到 1FFh。地址长度为 9 比特，每一地址位置可存储 8 位数据。

通过读 / 数据流读、编程和擦除操作来装入并修改 UFM 的数据。读 / 数据流读操作用于读取地址寄存器所指向地址的内容。递增地址寄存器（数据流读）可实现对存储器的连续读操作。编程操作用于把数据装入 UFM，而修改 UFM 内容时则采用擦除操作。但是，不能只擦除单个地址位置。擦除操作要么擦除整个 UFM (A2A1A0 = 111)，要么擦除地址 MSB 所指向的 UFM 区。

UFM 还含有一个可以使能的内部振荡器。通过逻辑阵列模块来传送其输出信号，还可以把该信号反馈回 ARCLK 和 DRCLK。

本设计使用 I²C 串口对 UFM 进行数据存储和读取。下面总结了 I²C 协议：

- 使用双向总线：SDA 线用于地址和数据传送，SCL 线用于 I²C 时钟。释放后，两条线都保持高电平（上拉）。

- 当 SCL 高电平时，SDA 线上高电平到低电平的转换是通信开始的起始条件。
- 然后，将从机地址从 SDA 上发送出去。一旦从机对地址应答后，开始数据传送。时钟高电平时，SDA 线上传送的数据必须保持稳定。
- 当 SCL 高电平时，SDA 线上低电平到高电平的转换是通信结束的停止条件。

表 1 说明了 UFM 模块中出现的信号。

表 1. 对 UFM 模块中信号的简要说明		注释 (1)
信号	说明	
DRDin	每个 DRCLK 脉冲出现时，将数据移入数据寄存器。	
DRCLK	控制数据从 DRDin 移到 DRDout，以及数据从 UFM 并行装入到数据寄存器中。	
DRSHFT	高电平：从 DRDin 移入 LSB，从 MSB 移到 DRDout。 低电平：把数据从 UFM 锁存至数据寄存器。	
ARDin	串行输入，存放存储器地址。	
ARCLK	控制存储器地址偏移，以及地址寄存器中地址的递增量。	
ARSHFT	高电平：地址从 ARDin 串行移入地址寄存器 低电平：递增地址寄存器中的地址	
PROGRAM	在这一信号的上升沿，来自数据寄存器的数据被写入地址寄存器指向的存储器。	
ERASE	在这一信号的上升沿，地址寄存器 MSB 指向的存储器区被擦除。	
OSC_ENA	该信号被用于使能 UFM 的内部振荡器。	
DRDout	数据寄存器输出。MSB 首先输出。	
BUSY	在编程或者擦除指令中，指示存储器忙。	
RTP_BUSY	如果使能了实时 ISP 功能，则需要该信号。	

表 1 的注释：

(1) 如果需要了解详细信息，请参考 *Altera MAX II 器件手册*，使用 MAX II 器件用户闪存一章的“UFM 接口信号”表。

闪存宏功能用于例化 UFM。它支持用户选择接口以及所需的存储器容量，还可以选择对 UFM 中的数据进行写保护，选择使能内部振荡器，将其信号传送到 CPLD 的端口，还可以设置 UFM 从地址的前四位 (A6-A3 位)。它还提供选项，连接电路板上其他的三个从地址位 (A2 至 A0)。

在 Quartus II 软件中例化 UFM 宏功能

按照以下步骤来例化 UFM 宏功能：

1. 在 Quartus II 软件中，打开工程，例化内部振荡器。

- 在 Tools 菜单中，单击 **MegaWizard Plug-In Manager**。出现 **MegaWizard Plug-In Manager [page 1]** 对话框。
- 选择 **Create a new custom megafunction variation**，单击 **Next**。出现 **MegaWizard Plug-In Manager [page 2a]** 对话框。表 2 介绍了 **MegaWizard Plug-In Manager [page 2a]** 对话框中的选项和设置。

表 2. *MegaWizard Plug-In Manager [page 2a] 选项*

选项	设置
Which device family will you be using?	选择 MAX II 。
Which megafunction would you like to customize?	单击 “+” 图标，展开 Memory Compiler ，选择 Flash Memory 。
Which type of output file do you want to create?	从 AHDL 、 VHDL 或者 Verilog HDL 中进行选择。
What name do you want for the output file?	输入您的文件名。

- 单击 **Next**。出现 **MegaWizard Plug-In Manager ALTUFM [page 3 of 5]** 对话框。表 3 列出了 ALTUFM 向导第 3 页中的选项和设置。

表 3. *MegaWizard Plug-In Manager [page 3 of 5] 选项*

选项	设置	其他选项
What is the interface protocol?	None	Use arclkena port (clock enable for arclk)
		Use drclkena port (clock enable for drclk)
	Parallel	What is the width of the address bus?
		What is the width of the data bus?
		Use the ‘osc’ (oscillator) output port
	Serial Peripheral Interface (SPI)	Base mode (uses 8 bit address and data)
		Extended mode (uses 16 bit address and data)
		Use the ‘osc’ (oscillator) output port
	I ² C	What is the MSB of the device address (in binary)?
		What is the size of the memory?
Use the ‘osc’ (oscillator) output port		
What is the access mode for the user flash memory?	Read/Write 或者 Read Only	—

5. 单击 **Next**。

- 如果您选择 **I2C**，将出现 ALTUFM 向导第 4 页。表 4 列出了 ALTUFM 向导第 4 页的选项和设置。

单击 **Next**。出现 ALTUFM 向导第 3 页。

选项	设置	其他选项
What is the write configuration for the I2C protocol?	Single byte write	—
	Page write	从 8 bytes、16 bytes、32 bytes 中选择
Write protect	Use the 'wp' (write protect) input	Write protect applies to the full memory
		Write protect applies only to the upper half of the memory
What erase method should be used in I2C protocol?	Device Slave Address Full Erase (3 LSBs are 111)	—
	Sector Erase Triggered by Byte Address (1)	Sector 0: Trigger erase when writing to binary address (MSB is always '0')
		Sector 1: Trigger erase when writing to binary address (MSB is always '1')
	Sector Erase Triggered by 'a2' bit	—
No Erase	—	

表 4 的注释：

(1) 只有当您选择了 What is the write configuration for the I2C protocol? 下的 Single byte write 时，才出现该选项。

- 如果选择了 **None**、**Parallel** 或者 **Serial Peripheral Interface (SPI)**，出现 ALTUFM 向导第 4 页。表 5 列出了 ALTUFM 向导第 3 页中的选项和设置。

选项	设置
Do you want to specify the initial content of the memory?	No, leave it blank
	Yes, use this file for the memory content
What is the oscillator frequency for the User Flash Memory? (for simulation only)	选择 3.33 MHz 或者 5.56 MHz

表 5. MegaWizard Plug-In Manager [page 4 of 6] 选项	
选项	设置
What is the erase time for the User Flash Memory? (for simulation only)	输入数值
What is the program time for the User Flash Memory? (for simulation only)	输入数值

- 单击 **Next**。出现 ALTUFM 向导第 5 页。
- 在 ALTUFM 向导第 5 页，如果您选择了该选项，可使用该网表的文件。灰色标记表示自动产生的文件，红色标记表示为可选文件。单击 **Next**。
- ALTUFM 向导第 6 页显示了要生成的文件类型列表。自动生成的变量文件含有您在 2a 页上指定语言编写的打包代码。在第 7 页，您可以指定要生成的其他文件类型。从 AHDL Include 文件 (<function name>.inc)、VHDL 组件声明文件 (<function name>.cmp)、Quartus II 符号文件 (<function name>.bsf)、Instantiation 模板文件 (<function name>.v) 和 Verilog HDL 黑盒文件 (<function name>_bb.v) 中进行选择。单击 **Finish**。

设计实现

您可以采用 EPM240 或者其他 MAX II CPLD 来实现该设计。这一设计实例中的 UFM 分配了 I²C 接口。在 I²C 总线环境中演示了对 MAX II UFM 的访问。实现过程包括使用本应用笔记附录的设计实例源代码，将 UFM 的 I²C 接口线分配给 MAX II 的 GPIO 等。采用 I²C 仿真器对 UFM 进行读写操作，该仿真器由 PC 并口生成，并通过硬件连接来生成 I²C 兼容 2 线总线。关于建立 I²C 环境的详细信息，请参考 Dallas 半导体公司的 Maxim 应用笔记 AN3230：

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

从下面的链接下载相应的免费软件：

http://files.dalsemi.com/system_extension/AppNotes/AN3315/ParD S2W.exe

使用这一实用程序以及并口及其接口硬件和 MAX II CPLD 进行通信，并提供 SDA 和 SCL 连接（需要 I²C 2 线系统来访问 UFM）。完成后，该设计使 I²C 主机能够访问 UFM，读写数据。该演示中的 I²C 主机是运行并口 I²C 软件的 PC 用户接口。UFM 是 I²C 从机。

表 6 列出了怎样在 MDN-B2 演示板上实现该设计实例。

信号	引脚
SCLK	39 引脚
SDA	40 引脚
a1	37 引脚
a2	38 引脚

在 Quartus II 软件的 **Device and Pin Options** 对话框中, 把未使用的引脚分配为 **input tri-stated**。Quartus II 软件的 Assignment Editor 用于使能 SCLK 和 SDA 引脚的 **Auto Open-Drain Pins** 功能。这些设置之后是编译过程。

请参考下面的演示说明 (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。
- 在编程过程启动前以及启动过程中, 保持演示板上 SW4 的按下状态不变。一旦完成, 关断电源, 拔下 JTAG 连接器。
- 在 PC 上建立并口驱动 I²C 环境:
 - 下载 Maxim 并口实用程序等软件工具实现和 I²C 协议从机进行通信。安装并口软件。这一实例使用了程序 **ParDS2W.exe**: http://files.dalsemi.com/system_extension/AppNotes/AN3315/ParDS2W.exe
 - 对于这一并口实用程序, 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, 可以从下面的链接下载: 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 378h 至 37Fh), 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/EC P Printer port (LPT)/Resources。
 - 如果并口被配置为其他类型而不是 ECP, PC 启动时, 在 BIOS 设置中把它改为 ECP。

- 在 Direct IO 控制面板中选择 **Security Tab**, 浏览 **ParDS2W.exe** 程序路径。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 软件狗。如果需要延长并口连接以靠近演示板, 则使用延长线。
 - 将 I²C 并口软件狗的 4 针插头插入演示板的 I²C 插座 (JP3), 插头的红色标记对上 JP3 插座的 pin#1。
 - 把 SW3 (MDN-B2 演示板上的 8 路 DIP 开关) 的开关 1 和 2 打到 ON 位置。
 - 打开 ParDS2W 程序, 为 PC 选择合适的并口地址 (配置 Direct IO 时可以看到), 把 **2-Wire Device Address** 设置为 00H。
 - 最后, 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。
-
- 利用并口实用程序的 **2-Wire Utility** 中的 **One byte Write/Read**, 指定地址和数据 (每个为一个字节长度, 2 个十六进制位), 对任意指定的存储器地址进行单字节 **WRITE** 操作。
 - 同样的, 在相同的地址位置进行 **READ** 操作, 注意该地址位置的内容。它应该是刚写入的数值。任何其他地址位置都应该是 FFh, 除非被写入了数据。
 - 如果向一个地址位置多次进行 **WRITE** 操作, 那么应该先进行擦除操作。
 - 选择 BEh 作为从地址, 执行写入 FFh 的 **WRITE** 操作, 擦除所有的内容。这样, 把 UFM 的内容恢复为 FF。
 - 您可以使用演示板上 SW3 (8 路 DIP 开关) 的开关 1 和开关 2 来设置 I²C 从地址 (a2, a1) 的第 5 位和第 6 位。在闪存宏功能例化期间, 这些都置为 0, I²C 从地址的前 4 MSB 置位 (这一例子为 1011 或者 Bh)。

源代码

该应用笔记的设计实例采用了 Verilog HDL 实现，成功地运行在 MDN-B2 演示板上。下面的链接提供源代码、测试台文件和完整的 Quartus II 工程：

www.altera.com/literature/an/an489_design_example.zip

结论

对于需要非易失存储器支持的各种逻辑方案，MAX II CPLD 提供独特的片内用户闪存，是非常好的选择。而且，MAX II 器件还具有低功耗、简洁迅速的上电和多电压特性，并提供内置内部振荡器，是最通用的可编程逻辑器件。

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 器件应用笔记：
[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 7 列出了本应用笔记的版本历史。

表 7. 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。

