

使用十六进制 (HEX) 文件和重启时序实现ADM1260编程

作者: Hossain Opal

简介

ADM1260有两个512字节的非易失、可电擦除只读程序存储 (EEPROM) 单元, 位于地址0xF800至地址0xFBFF。EEPROM用于永久存储数据, 这些数据在ADM1260掉电时不会丢失。每个EEPROM分成16页, 每页32个字节。

在ADM1260中, EEPROM可以分成六个部分, 如表1所示。

本应用笔记说明在系统应用中对ADM1260执行配置更新所需的步骤。ADM1166、ADM1168和ADM1169系列编程均使用相同的步骤。

擦除EEPROM

如果EEPROM包含配置和时序数据, 则在进行新配置和时序数据编程之前, 应擦除EEPROM页。

要擦除EEPROM空间, 请执行以下步骤:

1. 将0x05写入寄存器0x90, 以使能EEPROM数据块擦除。
2. 擦除或访问0xFA00至0xFBFF地址范围的EEPROM时, 首先应暂停时序控制引擎。将0x01写入SECTRL寄存器0x93, 以暂停时序控制引擎。每次对时序控制引擎的EEPROM空间执行读或写处理时, 都必须暂停该引擎。
3. 擦除或访问0xF800到0xF89F或0xF900到0xF9FF地址范围的EEPROM时, 应将0x01写入地址为0x9C的BBCTRL寄存器。这可以暂停黑匣子操作, 并使能对EEPROM的0xF800到0xF89F和0xF900到0xF9FF地址范围的访问。
4. 将0xF800写入ADM1260, 它是EEPROM A第1页的起始地址。

5. 将0xFE写入器件, 它擦除第0页。

6. 重复步骤4和步骤5, 每次迭代让低位地址偏移32。步骤4中的地址为0xF820、0xF840、0xF860, 依此类推。在后续的擦除操作之间添加25 ms延迟。

读取十六进制文件

使用ADM1260的配置生成的十六进制文件遵循Intel十六进制格式, 它将二进制信息转换为ASCII文本形式。

十六进制文件的每一行都包含了多个十六进制数, 表示数据以及存储的地址。

图1显示了一个ADM1260的典型记录。

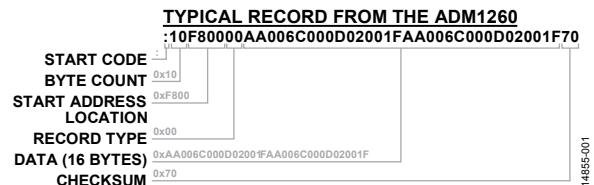


图1. 来自ADM1260的典型记录

将解码字节值相加, 计算每行数据的检验和, 如下所示:

$$10 + F8 + 00 + 00 + AA + 00 + 6C + 00 + 0D + 02 + 00 + 1F + AA + 00 + 6C + 00 + 0D + 02 + 00 + 1F = 0x390$$

解码字节值之和0x390, 与0xFF进行逻辑“与”运算, 得出的结果为0x90。0x90的二进制补码为0x70, 它是校验和。

十六进制文件的末尾以00000001FF表示。

表1. EEPROM地址映射

类型	起始地址 (十六进制)	结束地址 (十六进制)	EEPROM	页
配置数据	0xF800	0xF89F	A	0至4
保留	0xF8A0	0xF8FF	A	5至7
GUI信息 (仅适用于ADM1260)	0xF900	0xF93F	A	8至9
用户空间	0xF940	0xF97F	A	10至11
故障记录	0xF980	0xF9FF	A	12至15
时序数据	0xFA00	0xFBFF	B	0至15

将配置和时序数据写入EEPROM

要将配置和时序数据写入EEPROM，请执行以下步骤：

1. 将0x01写入寄存器0x90，使得配置寄存器能够持续更新。
2. 将0xF800写入器件，它是EEPROM A第1页的起始地址。
3. 使用命令0xFC，写入要发送的数据字节数，也就是0x20（32个字节）后跟32字节数据。
4. 重复步骤2和步骤3，每次迭代让低位地址偏移32。步骤2中的地址为0xF820、0xF840、0xF860，依此类推。

重新加载新配置并重启时序

将数据写入EEPROM之后，通过将0x01写入寄存器0xD8 (UDOWNLD)，可加载新配置。此写入可从EEPROM下载配置数据。

要从保留状态重启时序控制引擎，请连续将0x27和0x10写入地址为0xDA的寄存器UNLOCKSE，然后将0x01写入地址为0xDB的SEDOWNLD寄存器，从而解锁SEDOWNLD寄存器。

在多器件系统中，只需要重启某一个ADM1260器件，这会导致通过IDB连接的其他ADM1260器件重启。