

4. 32 位模式, 读 E2PROM

- 设置 EEARH/L 为目标地址, 地址为 4 字节对齐
- 设置 EEP[3:1] = 010, 开启 32 位接口模式
- 设置 EERE = 1, 启动 E2PROM 读操作
- 等待 2 个系统时钟周期 (执行两个 NOP 指令)

E2PCTL 访问模拟 E2PROM 空间, 支持连续编程模式, 连续访问模式对于需要一次更新一个数据块的应用非常高效, 也有利于提高 FLASH 的使用寿命。连续编程模式仅支持 32 位宽度的数据编程操作。

连续访问模式通过 ECCR 寄存器的 SWM 位使能。SWM 使能后, 接下来通过 E2PCTL 写模拟 E2PROM 空间的操作都在连续编程模式。在连续编程模式下, E2PCTL 控制器会根据目标地址内的数据情况自动处理换页。但在连续编程模式过程中如果发生换页, 控制器在连续编程过程中, 不会自动将 CP0/1 区域的数据交换, 也不会更新页面信息。

当连续编程到最后一次操作前, 通过清零 SWM 位关闭连续编程模式, 然后在非 SWM 模式下启动最后一次编程操作, 编程结束后, E2PCTL 会自动将 CP0/1 区域的数据复制到交换页, 并更新交换页的信息, 使之成为当前有效页, 从而完成整个连续编程操作。

5. 连续编程模式操作流程:

1. 通过 ECCR 配置数据 FLASH 的大小, 并使能 SWM 位
2. 使用 32 位模式编程模拟 E2PROM 区域
3. 如果不是最后一次操作, 回到步骤 2 继续编程下一个数据
4. 如果达到最后一次编程, 首先通过 SWM 禁止连续编程模式, 然后使用步骤 2 的操作流程完成最后一次编程

E2PCTL 高效 FLASH 数据管理

E2PCTL 控制器除了实现连续编程模式, 也可以通过 ECCR 寄存器的 CP0/1 位对页交换过程数据交换复制进行独立控制。ECCR 寄存器的 CP0/1 分别用于控制页交换过程中对于当前页面中 CP0/1 区域数据的交换操作。清零 CP0/1 位, 在页交换过程中将不会交换当前页中对应区域的数据。本节提供的一种高效管理方法, 将会利用这一特性。

在 FLASH 数据更新过程中, 最为耗时的操作发生在交换页擦除过程。因此我们可以寻址一种最大限度减小页擦除次数的数据管理方法, 既能提高编程效率, 也能减少寿命损耗。

这里我们提供一种参考算法, 适用于基于数据块数据管理应用:

1. 假定用户数据只是一个完整的数据块, 数据块大小 4 字节的整数倍;
2. 每次数据更新将会更新一个完整的数据块
3. 数据块信息除了存放用户数据, 还需要存放一个块管理信息

以上三个条件下, 我们可以充分利用 E2PCTL 的连续编程模式和自动页交换机制, 实现一个高效率的 FLASH 数据管理方法。

由于是每次更新的数据为一个相同大小的数据块, 并且每块数据结构中保存有指向下一块数据的地址信息, 因此我们可以每次更新数据时按地址顺序编程 FLASH, 无需做 CP0/1 的数据复制。同时由于每次都是更新数据到一个已擦除的区域, 也不会发生页擦除。

当数据写完最后一块, 其结构信息指向的下一块数据区回到页的起始地址。此后再发生数据写操作, E2PCTL 将会启动一次页擦除过程, 并更新当前活动页面。