

硬核处理器系统 (HPS) 提供一个 NAND 闪存控制器以与 Altera® 片上系统 (SoC) FPGA 系统中的外部 NAND 闪存存储器相连接。可以使用外部闪存存储器来存储一个处理器启动镜像、软件，或用来存储大型应用程序和用户数据。HPS NAND 闪存控制器基于 Cadence® Design IP® NAND 闪存存储器控制器。

NAND 闪存控制器功能

NAND 闪存控制器具有以下功能：

- 支持一个 x8 NAND 闪存器件
- 支持开放 NAND 闪存接口 (ONFI) 1.0
- 支持 Hynix、Samsung、Toshiba、Micron 和 ST Micro 的 NAND 闪存存储器
- 支持可编程的 512 字节 (4-、8- 或 16-bit 纠错) 或 1024 字节 (24-bit 纠错) 错误纠正编码 (ECC) 扇形容量
- 支持流水线预读和写命令以实现增强的读 / 写吞吐量
- 支持每块包含 32、64、128、256、384 或 512 页面的器件
- 支持多平面器件
- 支持 512 byte、2 kilobyte (KB)、4 KB 或 8 KB 的页面容量
- 支持可编程的纠正容量的单层单元 (SLC) 和多层单元 (MLC) 器件
- 提供内部直接存储器访问 (DMA)
- 提供可编程的访问时序

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

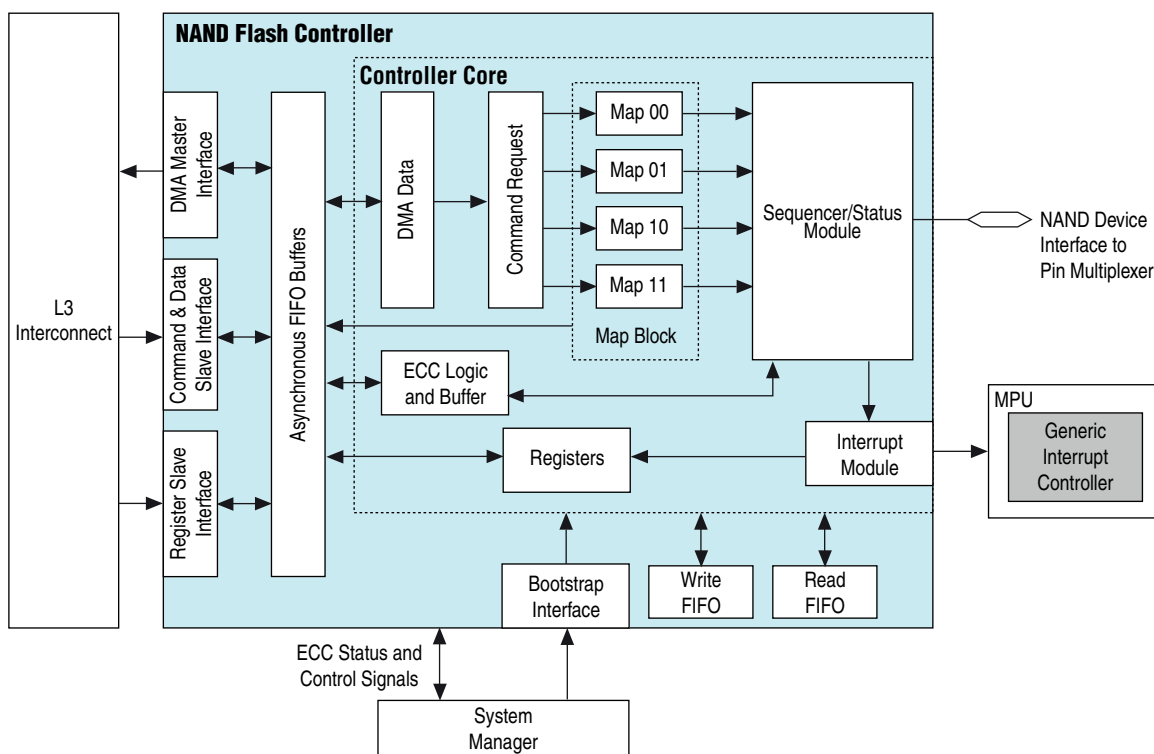
Portions © 2011 Cadence Design Systems, Inc. Used with permission. All rights reserved worldwide. Cadence and the Cadence logo are registered trademarks of Cadence Design Systems, Inc. All others are the property of their respective holders.



NAND 闪存控制器结构图和系统集成

图 10-1 显示了 HPS 中 NAND 闪存控制器的集成。闪存控制器通过命令和数据从接口从主机接收命令和数据。主机通过寄存器从接口访问闪存控制器的控制和状态寄存器 (CSR)。闪存控制器处理所有命令序列和闪存器件的协同工作。当从 NAND 闪存存储器启动 HPS 时，引导程序 (bootstrap) 接口支持 NAND 闪存控制器的配置。闪存控制器生成对 HPS Cortex™-A9 MPCore™处理器通用中断控制器的中断。DMA 主接口通过控制器的内置 DMA 提供往返闪存控制器的访问。

图 10-1. NAND 闪存控制器结构图



NAND 闪存控制器的功能说明

这一部分介绍 NAND 闪存控制器的功能性。

检测并初始化

HPS 上电并且等闪存器件稳定后，NAND 闪存控制器会按照一个指定的序列进行初始化。初始化期间，闪存控制器查询闪存器件并且根据以下两种闪存器件类型的其中之一进行配置：

- ONFI 1.0 兼容器件
- 传统（非 ONFI）NAND 器件

通过发送 Read Electronic Signature 命令，NAND 闪存控制器使用 ONFI 发现协议识别连接的 ONFI 兼容器件。对于不识别该命令的器件（尤其对于 512-byte 页面容量器件），软件必须在置低复位前写入系统管理器来置位 bootstrap_512B_device 信号以识别器件类型。

要支持启动和初始化，必须连接 rdy_busy_in 管脚。NAND 闪存控制器对连接的器件发出 reset 命令。

NAND 闪存控制器执行以下初始化步骤：

1. 如果系统管理器置位 bootstrap_inhibit_init，那么闪存控制器就会直接转到步骤 7。
2. 当器件准备好时，闪存控制器发送 ONFI Read ID 命令以读取器件的 ONFI 签名，以确定外部连接的器件是 ONFI 或者传统器件。
3. 如果器件返回的数据具有一个 ONFI 签名，那么闪存控制器读取器件参数页面。闪存控制器将相关器件功能信息存储在内部存储控制寄存器中，从而使其能够正确地编程闪存器件中的其它寄存器，并且转到步骤 5。
4. 如果数据没有一个有效 ONFI 签名，那么闪存控制器假设它是一个传统（非 ONFI）器件。闪存控制器会执行以下步骤：
 - a. 对器件发送 reset 命令
 - b. 读取器件签名信息
 - c. 将相关值存储到内部存储控制器寄存器
5. 闪存控制器复位器件。同时，它验证存储器接口的宽度。HPS 支持一个 8-bit NAND 闪存器件。因此，闪存控制器总是检测到一个 8-bit 存储器接口。
6. 闪存控制器发送 Page Load 命令到器件的页面 0 的模块 0，配置直接读访问，以便处理器可以从该页面启动。处理器可以开始从器件的第一个页面读取，它是预加载器软件的预期位置。



系统管理器可以在置低复位之前通过置位 bootstrap_inhibit_b0p0_load 旁路该步骤。

7. 闪存控制器将 reset 命令发送到器件。
8. 闪存控制器在 status 组的 intr_status0 寄存器中设置 rst_comp bit 的值。

引导程序接口

NAND 闪存控制器具有一个引导程序接口，使得软件可以覆盖闪存控制器的默认行为。引导程序接口包含 4 个 bit，当被正确地设置时能够使闪存控制器跳过初始化阶段并且在复位后立即开始从闪存存储器加载。这些 bit 通过系统管理器由软件驱动。当控制器从复位释放时，NAND 闪存控制器对它们进行采样。



要了解关于引导程序接口控制位的更多信息，请参考 *Cyclone V 器件手册* 第 3 卷的 *System Manager* 章节。

表 10 - 1 列出了 nandgrp 组的系统管理器的 bootstrap 寄存器中的相关引导程序设置位。该表也列出了对 512 字节页面器件所推荐的引导程序设置。

表 10 - 1. 对 512 字节页面器件所推荐的引导程序设置

寄存器	值
noinit	1 ⁽¹⁾
page512	1
noloadb0p0	1
tworowaddr	■ 1— 闪存器件支持 2 个周期的地址 ■ 0— 闪存器件支持 3 个周期的地址
表 10 - 1 注释： (1) 当该寄存器被设置时，NAND 闪存控制器期待主机编程相关的器件参数寄存器。要了解更多信息，请参考“主机进行的配置”。	

主机进行的配置

如果系统管理器将 bootstrap_inhibit_init 设置为 1，那么 NAND 闪存控制器不会执行发现和初始化过程。在这种情况下，主机处理器必须配置闪存控制器。

当在设计中不用考虑性能时，可以不需要更新时序寄存器。

表 10 - 2 显示了主机设置所推荐的配置以便对一个单平面、512 字节 / 页面器件使能基本的读、写和擦除操作。

表 10 - 2. 对 512 字节页面器件所推荐的引导程序设置

寄存器 ⁽¹⁾	值
devices_connected	1
device_width	0 表示一个 8-bit NAND 闪存器件
number_of_planes	1 表示一个单平面器件
device_main_area_size	该寄存器的值必须反映闪存器件的页面主要区域容量。
device_spare_area_size	该寄存器的值必须反映闪存器件的页面备用区域容量。
pages_per_block	该寄存器的值必须反映闪存器件中每模块页面的数量。

表 10 - 2 注释：
(1) 所有的寄存器都在 config 组中。

每个 NAND 页面都有一个主要区域和备用区域。主要区域用于数据存储。备用区域用于 ECC 和维护数据 (maintenance data)，例如耗散均衡 (wear leveling) 信息。每个模块包含一组页面。

主要和备用区域的容量，以及页面中模块的数量，取决于连接到 NAND 闪存控制器的指定的 NAND 器件。因此，器件相关的寄存器，必须编程 device_main_area_size、device_spare_area_size 和 pages_per_block 以匹配器件的特性。

如果软件不执行发现和初始化序列，那么软件必须包含一个替代方法以决定器件相关寄存器的正确值。HPS boot ROM 代码通过默认 (bootstrap_inhibit_init = 0) 使能发现和初始化。

时钟

表 10 - 3 列出了 NAND 闪存控制器时钟输入。

表 10 - 3. NAND 闪存控制器的时钟输入

时钟信号	说明
nand_x_clk	用于主和从接口以及 ECC 扇形缓冲器的时钟
nand_clk	用于 NAND 闪存控制器的时钟

nand_x_clk 的频率是 nand_clk 的频率的 4 倍。

要了解关于时钟输入的更多信息，请参考 *Cyclone V 器件手册* 第 3 卷的 *Clock Manager* 章节。

复位

NAND 闪存控制器具有一个复位信号。硬复位或者软复位时，复位管理器将该信号驱动到 NAND 闪存控制器。

NAND 闪存控制器退出复位状态之前，必须配置闪存外部接口的管脚多路复用器。

要了解关于复位管理器的更多信息，请参考 *Cyclone V 器件手册* 第 3 卷的 *Reset Manager* 章节。

索引寻址

NAND 闪存控制器使用索引寻址 (indexed addressing) 以减少闪存控制器消耗的地址跨度。间接寻址由两个寄存器控制，通过 nanddata 映射中的命令和数据从接口而被访问，如表 10 - 4 中所介绍。

表 10 - 4. 索引寻址的寄存器映射

寄存器名称	偏移地址	使用
Control	0x0	软件写入 32-bit 控制信息，包含 MAP 命令类型、模块和页面地址。上部 4 个位必须被设置为 0。要了解关于 Control 寄存器的具体使用的详细信息，请参考表 10 - 5 到表 10 - 8。
Data	0x10	Data 寄存器是一个到 NAND 闪存的页面容量的窗口。通过从该偏移开始的位置读取或写入，软件直接从 Control 寄存器指定的 NAND 闪存存储器的页面和模块读取或写入。

主机如下所示使用按索引寻址：

1. 将 32-bit 索引寻址域编程到数据 / 命令从端口的 Control 寄存器的偏移 0x0。这一操作对 NAND 闪存控制器提供闪存地址参数。
2. 在数据 / 命令从端口的 Data 寄存器的偏移 0x10 执行 32-bit 读或写
3. 如果它们在闪存存储器的相同的页面和模块中，那么执行额外的 32-bit 读和写

如果一组数据传输到相同的页面和模块地址，那么没有必要对每个数据传输写入控制寄存器。例如，可以使用模块和页面地址在页面开头写入控制寄存器，然后通过指导连续的传输到 Data 寄存器来读或写整个页面。

命令映射

NAND 闪存控制器支持几个闪存控制器指定的 MAP 命令，提供一个抽象层以对 NAND 闪存器件进行编程。通过使用 MAP 命令，可以避免直接对器件指定的命令进行编程。使用该抽象层可以提供增强的性能。命令要用多个周期以发送到片外。MAP 命令使您能够启动命令并且使闪存控制器将它们排序到片外的 NAND 器件。

NAND 闪存控制器支持以下闪存控制器指定的 MAP 命令：

- **MAP00 命令** —read-modify-write 操作期间的 boot-read 或缓冲器读 / 写
- **MAP01 命令** — 存储器阵列读 / 写
- **MAP10 命令** —NAND 闪存控制器命令
- **MAP11 命令** — 低级直接访问

MAP00 命令

MAP00 命令访问 NAND 闪存器件中的页面缓冲器。地址总是从 0x0 开始并且扩展到 config 组中 device_main_area_size 和 device_spare_area_size 寄存器指定的页面容量。可以使用该命令执行启动读 (boot read)。使用 read-modify-write (RMW) 操作中的 MAP00 命令以便读或写缓冲器中的所有字。MAP00 命令支持到器件的页面缓冲器的直接数据路径。

只有当其它的 MAP01 或 MAP10 命令在 NAND 闪存控制器中不工作时，主机才可以使用 MAP00 命令直接访问页面缓冲器。

表 10 - 5 列出了 NAND 闪存控制器解释的 MAP00 命令的地址位。

表 10 - 5. MAP00 地址映射

地址位	名称	说明
31:28	(保留)	设置为 0
27:26	CMD_MAP	设置为 0
25:13	(保留)	设置为 0
12:2	BUFF_ADDR	存储器器件的数据宽度对齐的缓冲器地址。最高页面访问是 8 KB。
1:0	(保留)	设置为 0

正常操作下该命令的使用限制于以下情况中：

- 它可用于从器件执行一个Execute-in-Place (XIP) 启动；从器件直接启动的同时从页面缓冲器直接读取。
- MAP00 命令可以用于执行 RMW 操作，其中 MAP00 写操作用于修改器件页面缓冲器中的读页面。因为 NAND 闪存控制器在此操作期间不执行 ECC 纠错，所以在 MLC 器件中不建议使用该方法。
- 和MAP11命令一起，MAP00命令提供一种方式使主机可以使用MAP01和MAP10命令旁路 NAND 闪存控制器提供的硬件抽象而直接访问器件。该方法也用于调试，或者用于发出闪存控制器可能不支持 MAP01 或 MAP10 命令的操作。

限制：

- MAP00 命令不可以和 MAP01 命令一起使用来读取页面的部分。使用 MAP01 命令的访问必须执行一个完整页面传输。
- ECC 不在 MAP00 数据访问期间被执行。
- 执行 MAP00 操作时必须禁用 DMA (dma 组中的 dma_enable 寄存器的 flag 位必须设置为 0)。

MAP01 命令

MAP01 命令传输主机存储器和 NAND 闪存器件的指定页面之间的完整页面。因为 NAND 闪存控制器仅支持页面地址，所以必须立即读或写整个页面。所需命令的实际数量取决于数据传输的容量。即使需要多个命令，整个页面的传输也必须使用相同地址。

当 NAND 闪存控制器接收到读命令时，它对器件发出一个数据读取操作，等待读取完成，然后返回读数据。必须从页面的开头到页面的结束读取读数据。必须从页面的开头到页面的结束写入写数据。

当 NAND 闪存控制器接收到传输确定的消息时，它发出命令以将数据写到器件上。闪存控制器忽略读和写命令的字节使能并且传输整个数据宽度。

表 10 - 6 列出了 NAND 闪存控制器解释的 MAP01 命令的地址位。

表 10 - 6. MAP01 地址映射

地址位	名称	说明
31:28	(保留)	设置为 0
27:26	CMD_MAP	设置为 1
25:24	(保留)	设置为 0
23: <M> (1)	BLK_ADDR	器件中的模块地址
(<M> - 1) : 0 (1)	PAGE_ADDR	器件中的页面地址

表 10 - 6 注释：

- (1) <M> 取决于器件中每模块的页面数。<M> = ceil (log2 (<device pages per block>))。因此，使用以下值：
- 每模块 32 个页面： <M>=5
 - 每模块 64 个页面： <M>=6
 - 每模块 128 个页面： <M>=7
 - 每模块 256 个页面： <M>=8
 - 每模块 384 个页面： <M>=9
 - 每模块 512 个页面： <M>=9

NAND 闪存控制器采用 ECC 现场纠错，从闪存控制器传输数据之前，可以在内部纠正从器件读取的数据。ECC 扇形缓冲器存储数据，而 ECC 引擎计算错误位置。

请按如下所示使用 MAP01 命令：

- 必须使用 MAP01 命令读取或写入一个完整页面。在该传输期间，每个主机的传输必须具有相同的模块和页面地址。NAND 闪存控制器在内部跟踪它所读取或写入多少数据。
- MAP00 命令不可以在使用 MAP01 命令之间被用于读或写页面。
- 主机直接执行 MAP01 操作时，必须禁用 DMA (dma 组中 dma_enable 寄存器的 flag 位必须设置为 0)。如果使能 DMA 时，主机对 NAND 闪存控制器发出 MAP01 命令，那么闪存控制器丢弃请求并且生成一个 unsup_cmd 中断。

MAP10 命令

MAP10 命令对 NAND 闪存控制器的控制平面提供接口。MAP10 命令控制闪存器件的特殊功能，例如擦除、锁定、解锁、回写式 (copy back) 和页面备用区域访问。该命令路径中传递的数据以 NAND 闪存控制器而不是闪存器件为目标。不同于其它的命令类型，与这些传输相关的数据（输入或输出）不影响闪存器件的内容。但是，该数据指定并且执行闪存控制器的全部命令。只有 Data 寄存器的低 16 位包含相关信息。

表 10-7 NAND 闪存控制器所解释的 MAP10 命令的地址位。

表 10-7. MAP10 地址映射

地址位	名称	说明
31:28	(保留)	设置为 0
27:26	CMD_MAP	设置为 2
25:24	(保留)	设置为 0
23:<M> (1)	BLK_ADDR	器件中的模块地址
(<M>-1):0 (1)	PAGE_ADDR	器件中的页面地址

表 10-7 注释:

- (1) <M> 取决于器件中每模块的页面数。<M> = ceil (log2 (<device pages per block>))。因此，请使用以下值：
- 每模块的 32 个页面: <M>=5
 - 每模块的 64 个页面: <M>=6
 - 每模块的 128 个页面: <M>=7
 - 每模块的 256 个页面: <M>=8
 - 每模块的 384 个页面: <M>=9
 - 每模块的 512 个页面: <M>=9

表 10-8 列出了 MAP10 命令定义的特殊功能。

表 10-8. MAP10 操作

命令	功能
0x01	对擦除设置模块地址并且启动操作
0x10	设置解锁起始地址
0x11	设置解锁结束地址并且启动解锁
0x21	启动对所有模块的锁定
0x31	启动对所有模块的锁紧
0x41	设置备用区域访问
0x42	设置默认区域访问
0x43	设置主要 + 备用区域访问
0x60	在 RMW 操作中将页面加载到缓冲器
0x61	对 RMW 操作中的页面缓冲器设置目的地址
0x62	在 RMW 操作中写入页面缓冲器
0x1000	设置复制源地址 (copy source address)
0x11<PP>	设置复制目的地址并且启动 <PP> 页面的拷贝
0x20<PP>	设置 <PP> 页面的流水线预读
0x21<PP>	设置 <PP> 页面的流水线写

按如下所示使用 MAP10 命令：

- MAP10 命令可以用于发出到器件的命令，例如擦除、回写式、锁定或解锁。
- MAP10 流水线命令也可用于从闪存器件读取或写入器件模块边界内连续的多个页面。主机必须首先发出一个 MAP10 流水线读或写命令，然后发出 MAP01 命令以进行实际数据传输。MAP10 流水线读或写命令指示 NAND 闪存控制器使用高性能命令（例如缓存或多平面），因为闪存控制器具有多个连续页面的内容要被读取。页面一定不可以跨越模块边界。如果穿越模块边界，那么闪存控制器生成不支持的命令（unsup_cmd）中断并且丢弃命令。
- 高达 4 个流水线读或写命令可以被发送到 NAND 闪存控制器。
- 当 NAND 闪存控制器执行 MAP10 流水线读或写命令时，必须禁用 DMA（dma 组中 dma_enable 寄存器的 flag 位必须设置为 0）。应为主机通过闪存控制器直接传输来自和到闪存器件的数据，所以必须禁用 DMA。

MAP11 命令

MAP11 命令提供对 NAND 闪存控制器的地址和控制周期的直接访问，从而使得软件使用 Command 和 Data 寄存器将命令直接发送到闪存器件。如果闪存器件支持一个标准闪存命令不支持的器件指定的命令，那么 MAP11 命令可用。它也可以用于低级调试。

MAP11 命令提供到闪存器件的直接控制路径。这些命令直接在 NAND 器件接口上执行命令、地址和数据读 / 写周期。命令、地址和写数据值位于 Data 寄存器。读数据时，返回数据也出现在 Data 寄存器中。间接地址寄存器编码控制操作类型。器件的命令和地址周期必须是主机总线的写传输。对于数据周期，主机总线（读 / 写）的传输类型决定器件接口的数据周期类型。使用 MAP11 命令时，主机仅可以发出对数据从端口的单拍访问。

表 10 - 9 列出了 NAND 闪存控制器所解释的 MAP11 命令的地址位。

表 10 - 9. MAP11 地址映射

地址位	名称	说明
31:28	(保留)	设置为 0
27:26	CMD_MAP	设置为 3
25:2	(保留)	设置为 0
1:0	TYPE	设置如下所示的控制器类型： <ul style="list-style-type: none">■ 0 = 命令周期■ 1 = 地址周期■ 2 = 数据读 / 写周期

按如下所示使用 MAP11 命令：

- 仅在特殊的情况中使用 MAP11 命令，以便调试或发送 NAND 闪存控制器不支持的器件指定的命令。
- 使用 MAP11 操作之前必须禁用 DMA。
- 使用 MAP11 命令时，主机仅可以使用单拍访问传输。



MAP11 命令提供对 NAND 闪存器件直接的无结构化访问。错误的使用会导致不可预测行为。

数据 DMA

DMA 传输数据需要很少的主机干预。软件使用 MAP10 命令启动数据 DMA。

dma 组中 dma_enable 寄存器的 flag 位使能数据 DMA 功能性。只有当工作的传输不在 NAND 闪存控制器中待定时才使能或禁用该功能性。使能 DMA 时，闪存控制器通过 DMA 主接口进行每 MAP10 命令启动一个 DMA 的传输。当禁用 DMA 时，闪存控制器的所有操作通过数据 / 命令从接口发生。

NAND 闪存控制器支持高达 4 个未决的 DMA 命令，并且忽略额外的 DMA 命令。如果软件发出多于 4 个未决的 DMA 命令，那么闪存控制器发出 unsup_cmd 中断。一收到 DMA 命令，闪存控制器就会执行命令序列以传输 DMA 命令中所要求的页面数。DMA 主器件从系统内存中的编程了的突发长度块中读或写页面数据。DMA 命令完成后，闪存控制器就会发出一个中断，并且开始执行序列中的下一个 DMA 命令。

流水线操作使得 NAND 闪存控制器执行相同类型的背靠背命令时能够优化性能。

在某些限制下，闪存控制器执行 DMA 传输时，非 DMA MAP10 命令可以被发送到 NAND 闪存控制器。MAP00、MAP01 和 MAP11 命令不可以在 DMA 模式被使能时被发送，因为闪存控制器在非常紧耦合的高性能数据传输模式中操作。一接收到错误命令 (MAP00、MAP01 或 MAP11)，闪存控制器就会发出一个 unsup_cmd 中断以通知主机违规命令。

如果使能数据 DMA (dma 组中 dma_enable 寄存器的 flag 位被设置为 1)，那么当主机发送一个数据 DMA 命令时，NAND 闪存控制器在闪存器件和主机存储器之间传输数据。传输一完成，闪存控制器就会通过置位中断通知主机。

- 数据 DMA 命令是一种 MAP10 命令。该命令由数据 DMA 引擎而不是由闪存控制器内核解释。
- 使能 DMA 时不支持 MAP01、MAP00 或 MAP11 命令。
- 闪存控制器接受数据 DMA 命令之前，必须通过设置 dma 组中 dma_enable 寄存器的 flag 位使能 DMA。
- 当 DMA 被使能并且 DMA 引擎启动数据传输时，可以使能 ECC 以便进行数据传输时的所需的数据校正。
- MAP10 命令和与 MAP01 命令相似的数据传输一起被使用。
- 除了数据 DMA 命令以及 MAP10 流水线读和写命令以外，所有其它 MAP10 命令 (例如擦除、锁定、解锁和回写式) 都被转送到闪存控制器。
- 闪存控制器随时都可以处理高达 4 个未决的数据 DMA 命令。多页面操作期间，DMA 传输一定不可以跨越闪存模块边界。如果跨越，那么闪存控制器会生成一个不支持的命令 (unsup_cmd) 中断并且丢弃命令。
- 数据 DMA 命令通常为多页面读和写命令，带有一个相关的主机存储器指针。多页面数据传输到主机存储器指针或从主机存储器指针开始传输。
- 数据 DMA 使用 dma 组中的 flash_burst_length 寄存器以决定在互联上驱动的突发长度值。数据 DMA 硬件不考虑互联的边界跨越限制。主句必须初始化起始主机地址，以便 DMA 主接口突发不会跨越 4 KB 边界。

两个方法可以启动 DMA 传输：多传输 DMA 命令和突发 DMA 命令。

多传输 DMA 命令

要使用多传输 DMA 命令启动 DMA，应该发送如表 10-10 到表 10-13 所示的 4 个命令-数据对到 NAND 闪存控制器的数据和控制从端口。

只有当 NAND 闪存控制器按顺序接收到 4 个命令-数据对时，它才处理多传输 DMA 命令。闪存控制器使用 unsup_cmd 中断对无序命令做出响应。如果排序的命令与其它的闪存控制器 MAP 命令交错，那么闪存控制器也使用 unsup_cmd 中断做出响应。

表 10 - 10 到表 10 - 13 显示了每个命令-数据对的格式。

表 10 - 10. 命令-数据对 1

	31:28	27:26	25:24	23:<M> (1)	(<M> - 1):0 (1)
命令	0x0	0x2	0x0	模块地址	页面地址

	31:16	15:12	11:8	7:0
数据	0x0	0x2	0x0 = 读 0x1 = 写	<PP>= 页面数

表 10 - 10 注释：

- (1) <M> 取决于器件中每模块的页面数。<M> = ceil (log2 (<device pages per block>))。因此，请使用以下值：
- 每模块的 32 个页面：<M>=5
 - 每模块的 64 个页面：<M>=6
 - 每模块的 128 个页面：<M>=7
 - 每模块的 256 个页面：<M>=8
 - 每模块的 384 个页面：<M>=9
 - 每模块的 512 个页面：<M>=9

表 10 - 11. 命令-数据对 2

	31:28	27:26	25:24	23:8	7:0
命令	0x0	0x2	0x0	存储器地址高位 (1)	0x0

	31:16	15:12	11:8	7:0
数据	0x0	0x2	0x2	0x0

表 10 - 11 注释：

(1) 主机存储器中的缓冲地址，它必须与 32 位对齐。

表 10 - 12. 命令-数据对 3

	31:28	27:26	25:24	23:8	7:0
命令	0x0	0x2	0x0	存储器地址低位 ⁽¹⁾	0x0

	31:16	15:12	11:8	7:0
数据	0x0	0x2	0x3	0x0

表 10 - 12 注释：

(1) 主机存储器中的缓冲地址，它必须与 32 位对齐。

表 10 - 13. 命令-数据对 4

	31:28	27:26	25:24	23:17	16	15:8	7:0
命令	0x0	0x2	0x0	0x0	INT 突发长度		0x0

	31:16	15:12	11:8	7:0
数据	0x0	0x2	0x4	0x0

表 10 - 13 注释：


(1) DMA 传输结束时，INT 指定生成主机中断。DMA 传输结束时，INT 控制 status 组中 intr_status0 寄存器的 dma_cmd_comp 位的值。INT 可以显示以下值的其中之一：

0— 不中断主机。dma_cmd_comp 位被设置为 0。

1— 中断主机。dma_cmd_comp 位被设置为 1。

要了解更多信息，请参考第 10 - 5 页的“索引寻址”。

如果想要 NAND 闪存控制器 DMA 执行可缓存的访问，那么必须通过在系统管理器中写入 nandgrp 组的 l3master 寄存器配置缓存位。使用系统管理器更改 NAND 闪存控制器 DMA 的缓存性能之前，它必须处于闲置状态。

 要了解关于系统管理器的更多信息，请参考 *Cyclone V 器件手册* 第 3 卷的 *System Manager* 章节。

当 NAND 闪存控制器处于 DMA 模式时，可以发出非 DMA MAP10 命令。例如，您可能在 DMA 命令之间触发一个主机启动的页面移动，以实现耗散均衡。然而，请不要在一组多传输 DMA 命令中的命令 - 数据对之间交错非 DMA MAP10 命令。必须在发送不同的命令前，发送表 10 - 10 到表 10 - 13 所示的所有 4 个命令 - 数据对。

当使能 DMA 时，不要发出 MAP00、MAP01 或 MAP11 命令。

使用多传输格式的 MAP10 命令被写入 nanddata 中 Data 寄存器的偏移 0x10，与使用递增 4 (INCR4) 格式的 MAP10 命令相同（在“突发 DMA 命令”中有所介绍）。

突发 DMA 命令

通过将 4 个 16-bit 访问的突发传输命令发送到 NAND 闪存控制器可以启动一个 DMA 传输。该 DMA 命令的形式可用于从 FPGA 架构中定制的 IP 启动 DMA 传输。大部分处理器内核不能使用该 DMA 命令的形式，因为它们不能够控制突发的宽度。

当使能 DMA 时，NAND 闪存控制器将 MAP10 流水线 DMA 命令 (使用表 10-14 显示的格式) 识别为 INCR4 命令。MAP10 流水线 DMA 命令的地址解码保持不变，如第 10-8 页的表 10-7 所示。表 10-14 列出了 MAP10 突发 DMA 命令结构。突发 DMA 命令以不同的格式运载和多传输 DMA 命令-数据对相同的信息。

使用 INCR4 格式的 MAP10 命令被写入 nanddata 中的 Data 寄存器的偏移 0x10，与使用多传输格式的 MAP10 命令相同 (在“多传输 DMA 命令”中有所介绍)。

表 10-14. MAP10 突发 DMA (INCR4) 命令结构

数据 Beat	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Beat 0	0x2				0x0: 读, 0x1: 写				<PP>= 页面数							
Beat 1 (1)	存储器地址高位															
Beat 2 (1)	存储器地址低位															
Beat 3	0x0							INT (2)	突发长度							

表 10-14 注释:

- (1) 主机存储器中的缓存地址，必须与 32 位对齐。
- (2) INT 在 DMA 传输结束时指定生成主机中断。INT 在 DMA 传输结束时控制 status 组中 intr_status0 寄存器的 dma_cmd_comp 位的值。
INT 可以显示以下值的其中之一：
0— 不中断主机。dma_cmd_comp 位被设置为 0。
1— 中断主句。dma_cmd_comp 位被设置为 1。

可以有选择地将表 10-14 中的 16-bit 域 (按顺序排列的 4 个单独的突发长度 1) 发送到 NAND 闪存控制器。Altera 推荐此方法。

如果想要 NAND 闪存控制器 DMA 执行缓存的访问，那么必须通过在系统管理器中写入 nandgrp 组的 l3master 寄存器配置缓存位。使用系统管理器修改 NAND 闪存控制器 DMA 的缓存性能之前，它必须处于闲置状态。



要了解关于系统管理器的更多信息，请参考 Cyclone V 器件手册第 3 卷的 *System Manager* 章节。

ECC

NAND 闪存控制采用 ECC 逻辑计算和纠正位错误。闪存控制器使用 Bose-Chaudhuri-Hocquenghem (BCH) 算法检测页中的多个错误。

NAND 闪存控制器支持 512- 和 1024- byte ECC 扇形。根据所选的的扇容量，闪存控制器对每个 512 或 1024 byte 的数据插入 ECC 检查位。在 512 或 1024 byte 后，闪存控制器将 ECC 检查位信息写入到器件页面。

ECC 信息在页面中数据的 512 或 1024 byte 之间被剥离。NAND 闪存控制器以相同的码型读取 ECC 信息并且出现的错误 512 或 1024 byte 的数据读取而计算。

表 10 - 15 列出了写入到备用区域的不同的纠错性能、扇形容量和所需检查位容量之间的关系。

表 10 - 15. 纠错性能、扇形容量和检查位容量

纠错	以字节为单位的扇形容量	以字节为单位的检查位容量
4	512	8
8	512	14
16	512	26
24	1024	46

NAND 闪存控制器提供了软件用来格式化页面的以下 ECC 编程模式：

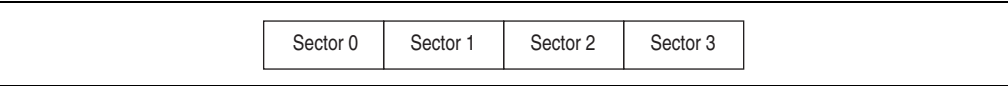
- 主要区域传输模式
- 备用区域传输模式
- 主要 + 备用区域传输模式

主要区域传输模式

在主要区域传输模式中，当 ECC 使能时，NAND 闪存控制器写时在数据流中插入 ECC 检查位并且读时剥离 ECC 检查位。当写入页面时，软件不需要管理 ECC 扇形。ECC 检查由闪存控制器执行，所以软件只传输数据。

如果 ECC 被关闭，那么 NAND 闪存控制器不读或写 ECC 检查位。图 10 - 2 显示了主要区域传输模式编程模型。

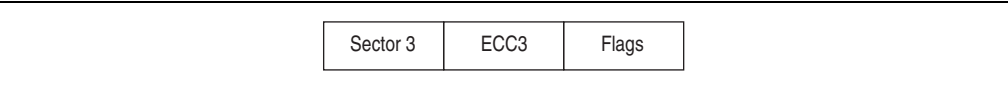
图 10 - 2. ECC 的主要区域传输模式编程模型



备用区域传输模式

NAND 闪存控制器不在备用区域传输模式中引入或解释 ECC 检查位，而让数据直接通过。图 10 - 3 显示了备用区域传输模式编程模型。

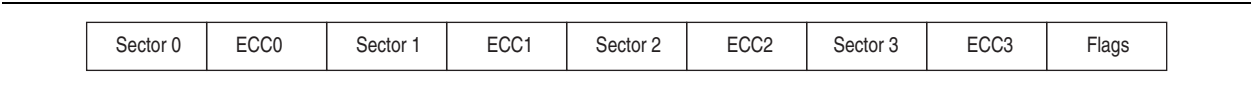
图 10 - 3. ECC 的备用区域传输模式编程模型



主要 + 备用区域传输模式

在主要 + 备用区域传输模式中，NAND 闪存控制器期待软件如图 10 - 4 所示格式化页面。当在写操作期间使能 ECC 时，闪存控制器生成的 ECC 检查位代替软件提供的 ECC 检查位数据。在读操作期间，闪存控制器将器件的 ECC 检查位转送到主机。如果禁用 ECC，那么从软件接收到的页面数据被写入到器件，并且从器件接收到的读数据被转送到主机。

图 10 - 4. ECC 的主要 + 备用区域传输模式编程模型



保存坏模块标记

当闪存器件生产商在生产过程中测试器件时，他们对所有坏的器件模块做出标记。每个坏模块被标记在指定的已知偏移，通常在备用区域的底部。一个坏模块标记是除 0xFF 之外的值（擦去的闪存的正常状态）。

使能 ECC 时，坏模块标记可以由页面中的最后扇区数据覆盖。这是因为 NAND 闪存控制器也使用页面的主要区域存储 ECC 信息，从而导致最后扇区扩展到备用区域。系统在写数据之前保存坏模块信息是有必要的，以便确保在闪存器件中正确识别坏模块。

当 NAND 闪存控制器将页面中的最后扇区写入到备用区域时，可以对它进行配置以使其跳过指定的字节数量。这个选项使得闪存控制器能够保存坏模块标记。要使用这个选项，将所需的偏移写入到 config 组中的 spare_area_skip_bytes 寄存器。例如，如果器件页面容量是 2 KB，并且器件生产商将坏模块标记存储在备用区域的前两个字节中，那么将 spare_area_skip_bytes 寄存器设置为 2。当闪存控制器写入与备用区域重叠的页面的最后扇区时，它从备用区域的偏移 2 开始，跳过偏移 0 的坏模块标记。值为 0（默认）表示没有跳过任何字节。spare_area_skip_bytes 的值必须是一个偶数。例如，如果坏模块标记为一个单字节，那么将 spare_area_skip_bytes 设置为 2。

在主要区域传输模式中，NAND 闪存控制器不会跳过坏模块标记。相反，它使用 config 组中 spare_area_marker 寄存器中编程的值覆盖坏模块标记。这个 8-bit 寄存器和 config 组中的 spare_area_skip_bytes 寄存器同时被使用以决定页面的备用区域中的哪个字节应该使用一个新标记值而被写入。例如，要标记一个模块为良好，需要设置 spare_area_marker 寄存器为 0xFF 并且设置 spare_area_skip_bytes 寄存器为标记应该被写入的字节的数量（从备用区域的底部开始）。

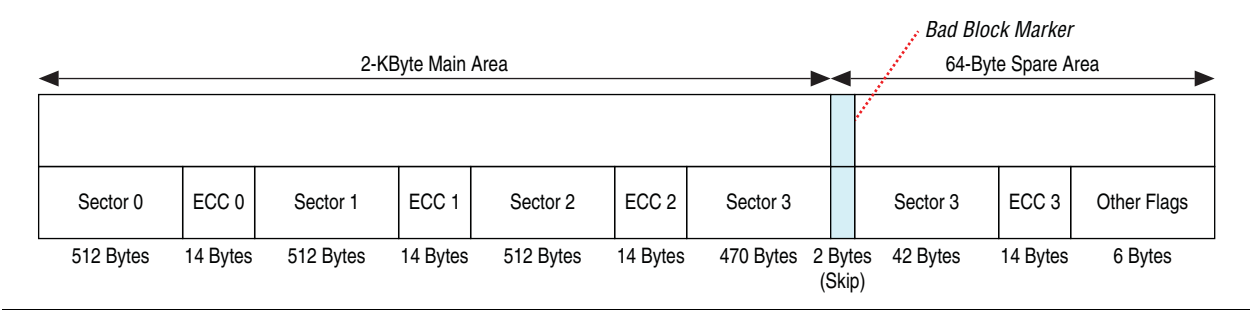
在备用区域传输模式中，NAND 闪存控制器忽略 spare_area_skip_bytes 和 spare_area_marker 寄存器。闪存控制器传输完全从主机或器件接收到的数据。

在主要 + 备用区域传输模式中，NAND 闪存控制器开始将页面的最后扇区写入到备用区域（从 spare_area_skip_bytes 寄存器中指定的偏移开始）。然而，包含坏模块识别信息的区域由主机写入到页面的数据覆盖。主机写入数据扇区和坏模块标记。闪存控制器根据主机软件在写入数据之前正确地建立坏模块标记。

要了解关于格式化该数据的更多信息，请参考第 10 - 14 页的图 10 - 4。

图 10 - 5 显示了 NAND 闪存控制器如何跳过坏模块标记的实例。该实例中，闪存器件具有 2-KB 页面以及 64-byte 备用区域。一个 14-byte 扇区 ECC 被显示，具有每扇区 8 字节纠错。

图 10 - 5. 坏模块标记



要了解在默认、备用或主要 + 备用区域传输模式中配置 NAND 闪存控制器的详细信息，请参考第 10 - 21 页的“传输模式操作”。

错误纠正状态

ecc 组中的 ECC 错误纠正信息 (ECCCorInfo_b01) 寄存器，包含 NAND 闪存控制器执行的每个读或写的错误纠正信息。ECCCorInfo_b01 寄存器在 max_errors_b0 和 uncor_err_b0 域中包含 ECC 错误纠正信息。

对进行中的传输的数据纠错结束时，ECCCorInfo_b01 保持最高数量的纠错应用于传输中的所有 ECC 扇区。除此之外，该寄存器表明整个传输是否具有可纠正的错误、不可纠正的错误或者无错误。当传输中的所有 ECC 扇区没有任何错误时，传输就没有错误。如果任何一个扇区都是不可纠正时，那么该传输被标记为不可纠正的。如果所有扇区都具有可纠正的错误并且没有扇区是不可纠正时，那么传输被标记为可纠正的。

每个传输结束时，主机必须读取该寄存器。该寄存器的值提供给主机关于模块的数据。所出现的可纠正的错误数量达到特定的阈值后，主机可以采取纠正操作。

接口信号

表 10 - 16 列出了用于 NAND 闪存接口信号的 I/O 管脚。

表 10 - 16. NAND 闪存接口信号

信号	宽度	I/O	说明
ad	8	in/out	闪存器件的命令、地址和数据
ale	1	out	地址锁存使能
ce_n	1	out	输出低电平有效芯片使能
cle	1	out	命令锁存使能
re_n	1	out	低电平有效读使能信号
rb	1	in	Ready/busy 信号
we_n	1	out	低电平有效写使能信号
wp_n	1	out	低电平有效写保护信号

HPS I/O 管脚支持一个单 x8 器件。

NAND 闪存控制器编程模型

该部分介绍 NAND 闪存控制器如何被运行在微处理器单元 (MPU) 上的软件编程。



如果您写入一个配置寄存器并且随后进行取决于该配置寄存器值的数据操作，那么 Altera 建议执行数据操作之前读取寄存器的值。该读取操作确保寄存器的后置写入完成并且生效，然后数据操作才发送到 NAND 闪存控制器。

基本闪存编程

该部分介绍软件访问和控制 NAND 闪存控制器必须采取的步骤。

NAND 闪存控制器优化序列

通过使用 config 组中 rb_pin_enabled 寄存器的 bank0 位，软件必须将闪存器件配置为中断或轮询模式 (polling mode)。如果器件处于轮询模式，那么软件同样必须编程额外的寄存器，以便选择轮询的次数和频率。请编程以下 config 组中的寄存器：

- 对于每个闪存器件，请将 rb_pin_enabled 寄存器设置为所需的操作模式。

- 对于轮询模式，将load_wait_cnt寄存器设置为相应的值(根据NAND闪存控制器的操作速度)，以及所需的等待值。
- 对于轮询模式，通过软件将program_wait_cnt寄存器设置为相应的值(根据NAND闪存控制器的操作速度)，以及所需的等待值。
- 对于轮询模式，通过软件将erase_wait_cnt寄存器设置为相应的值(根据NAND闪存控制器的操作速度)，以及所需的等待值。
- 对于轮询模式，通过软件将int_mon_cycnt寄存器设置为相应的值(根据NAND闪存控制器的操作速度)以及所需等待值。

通过使用 rb_pin_enabled 寄存器的 bank0 位，软件随时都可以将闪存器件从中断模式更改为轮询模式或反之亦然。

软件需要确保对该特定的闪存器件更改操作模式之前，该特定的闪存器件没有任何未决的传输。

器件初始化序列

初始化时，主机软件必须编程以下的 config 组中的寄存器：

- 将 devices_connected 寄存器设置为 1。
- 将 device_width 寄存器设置为 8。
- 将 device_main_area_size 寄存器设置为相应的值。
- 将 device_spare_area_size 寄存器设置为相应的值。
- 根据闪存器件的参数设置 pages_per_block 寄存器。
- 根据闪存器件的参数设置 number_of_planes 寄存器。
- 如果器件支持两个 ROW 地址周期，那么 two_row_addr_cycles 寄存器的 flag 位必须设置为 1。主机编程可以通过以下方法的其中之一确保实现该情况：
 - 在 NAND 闪存控制器的复位初始化序列之前将 bootstrap_two_row_addr_cycles 寄存器的 flag 位设置为 1，会导致闪存控制器自动初始化位。
 - 直接将 two_row_addr_cycles 寄存器的 flag 位设置为 1。
- 将 config 组中的 chip_enable_dont_care 寄存器清零。

NAND 闪存控制器可以识别闪存器件功能，使您能够初始化闪存控制器寄存器以正确地与器件连接，如第 10-2 页的“检测并初始化”中所介绍。

然而，少数 NAND 器件不遵循任何统一接受的识别协议。如果连接到这样的器件上，那么 NAND 闪存控制器不能够正确地识别它。如果使用这样的器件，那么软件必须使用其它的方法确保初始化寄存器被正确地设置。

器件操作控制

该部分提供了寄存器的列表，当在器件上选择使用多平面或缓存操作时需要它们进行编程。如果器件不支持多平面操作或缓存操作，那么这些寄存器可以保持其上电复位值而不会影响 NAND 闪存控制器的功能性。即使器件支持这些序列，软件也可能选择不用这些序列并且使这些寄存器仍处于其上电复位值。

编程以下 config 组中的寄存器以实现给定器件的最佳性能：

- 如果器件支持多平面操作访问与 NAND 闪存控制器连接的闪存器件的数据，那么将 config 组中 multiplane_operation 寄存器的 flag 位设置为 1。如果闪存控制器被设置为多平面操作，那么访问的页面数通常是器件中平面数的倍数。
- 如果 NAND 闪存控制器被配置为多平面操作，并且如果器件支持多平面读命令序列，那么设置 config 组中的 multiplane_read_enable 寄存器。
- 如果器件实现多平面地址限制，那么将 multiplane_addr_restrict 寄存器中的 flag 位设置为 1。
- 根据每器件要求，初始化 die_mask 和 first_block_of_next_plane 寄存器。
- 如果器件支持缓存命令序列，那么使能 config 组中的 cache_write_enable 和 cache_read_enable 寄存器。
- 如果器件不支持 copyback 命令序列，那么将 config 组中的 copyback_disable 寄存器的 flag 位清零。寄存器默认为使能状态。
- 因为 NAND 闪存控制器能够使用正确的序列（根据 NAND 闪存控制器的一些多平面或缓存相关的设置和生产商 ID 的组合），所以当前 config 组中的 read_mode、write_mode 和 copyback_mode 寄存器不需要由软件写入。如果将来这些设置更改，那么编程寄存器以适应更改。

ECC 使能

开始闪存器件的数据操作之前，需要决定想要使 ECC 使能或禁用。如果需要使能 ECC，那么根据器件的页面容量和可用备用区域设置相应的纠正情况。

将 config 组的 ecc_enable 寄存器中的 flag 位设置为 1 以便使能 ECC。如果被使能，那么 config 组中的以下寄存器需要被相应地编程，否则可以忽略它们。

- 初始化 ecc_correction 寄存器为相应的纠正情况。
- 如果软件需要保存坏模块标记，那么编程 config 组中的 spare_area_skip_bytes 和 spare_area_marker 寄存器。

要了解关于 ECC 的详细信息，请参考第 10 - 13 页的“ECC”。

NAND 闪存控制器性能寄存器

这些寄存器指定器件接口的突发容量，可以最大化 NAND 闪存控制器的整个性能。

通过最小化传输页面所需的突发数，初始化 dma 组中的 flash_burst_length 寄存器为一个最大化器件接口性能的值。

中断和 DMA 使能

启动 NAND 闪存控制器的数据操作之前，软件必须设置相应的中断状态寄存器位。如果软件选择使用闪存控制器中的 DMA 逻辑，那么必须设置寄存器空间中的相应的 DMA 使能和中断位。

- 将 config 组的 global_int_enable 寄存器中的 flag 位设置为 1，以便使能全局中断。
- 如果闪存控制器处于中断模式，那么发送任何操作之前将 status 组中的 intr_en0 寄存器的相关位设置为 1。
- 如果应用需要 DMA 模式，那么使能 DMA。通过设置 dma 组中 dma_enable 寄存器的 flag 位使能 DMA。Altera 建议软件读回该寄存器以确保发送 DMA 命令到闪存控制器之前，模式更改。

- 如果使能了 DMA，那么设置 dma 组中 dma_intr_en 寄存器的相应位。

中断状态位置位的顺序

以下中断状态位（在状态组的 intr_status0 寄存器中）以中断位设置的顺序列出：

1. time_out—当置位看门狗 time_out 位时，所有其它的中断位都被设置为 0。
2. dma_cmd_comp—DMA 操作传输数据期间，该中断状态位最后被置位。该位表示数据传输序列的完成。
3. pipe_cpybck_cmd_comp—当 copyback 命令或流水线命令的最后页面完成时，该位被置位。
4. locked_blk—当在锁定模块上执行编程（或擦除）时，该位被置位。
5. INT_act—与其它的中断状态位无关。表明对于该闪存器件，ready_busy 管脚值从 0 到 1 的跳变。
6. rst_comp—与其它的中断状态位无关。在复位命令完成后发生。
7. 对于一个擦除命令：
 - a. erase_fail（如果失败）
 - b. erase_comp
8. 对于一个编程命令：
 - a. locked_blk（如果在锁定模块上执行）
 - b. pipe_cmd_err（如果流水线序列由 MAP01 命令损坏）
 - c. page_xfer_inc（每个页面数据传输结束时）
 - d. program_fail（如果失败）
 - e. pipe_cpybck_cmd_comp
 - f. program_comp
 - g. dma_cmd_comp（如果使能 DMA）
9. 对于一个读命令：
 - a. pipe_cmd_err（如果流水线序列由 MAP01 命令损坏）
 - b. page_xfer_inc（每个页面数据传输结束时）
 - c. pipe_cpybck_cmd_comp
 - d. load_comp
 - e. ecc_uncor_error（如果失败）
 - f. dma_cmd_comp（如果使能 DMA）

时序寄存器

对于闪存器件的速度等级和时钟频率，必须优化以下寄存器。NAND 闪存控制器使用上电复位值正确地操作。然而，使用上电复位值运行是提供宽松时序（信号的大量裕量）的非优化模式。

对于连接的器件的速度等级和闪存控制器的操作频率，设置 config 组中的以下寄存器以优化 NAND 闪存控制器：

- twhr2_and_we_2_re
- tcwaw_and_addr_2_data
- re_2_we
- acc_clks
- rdwr_en_lo_cnt
- rdwr_en_hi_cnt
- max_rd_delay
- cs_setup_cnt
- re_2_re

忽略的寄存器

您不需要初始化以下 config 组中的寄存器：

- transfer_spare_reg 寄存器。可以使用 MAP10 命令初始化数据传输模式。
- write_protect 寄存器不需要被初始化（除非您正在检测写保护功能）。

闪存相关的特殊功能操作

该部分介绍可以在闪存存储器上执行的所有特殊功能。该功能由第 10-6 页的“命令映射”中介绍的 MAP10 命令所定义。

擦除操作

数据写入到闪存之前，需要执行一个擦除操作。NAND 闪存存储控制器支持单模块和多平面擦除。控制器从第 10-8 页的表 10-7 所示的间接地址解码模块地址。

单模块擦除

需要一个单命令以完成单模块擦除，如下所示：

1. 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为所需的擦除模块。
2. 将 0x01 写入 Data 寄存器。

对于单模块擦除，必须复位 config 组中的寄存器 multiplane_operation。

器件完成擦除操作后，控制器生成一个 erase_comp 中断。如果擦除操作失败，那么 erase_fail 中断被发出。失败模块的地址在 status 组的 err_block_addr0 寄存器中被更新。

多平面擦除

对于多平面擦除，config 组中的 number_of_planes 寄存器保持闪存器件中的平面数，并且指定的模块地址必须与器件中的平面数对齐。NAND 闪存控制器连续地擦除高达平面数的存储器的每个模块。请按如下所示发出该命令：

1. 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为所需擦除模块。
2. 将 0x01 写入 Data 寄存器。

对于多平面擦除，必须设置 config 组中的寄存器 multiplane_operation。

器件完成所有平面上的擦除操作后，NAND 闪存控制器生成一个 `erase_comp` 中断。如果擦除操作在多平面擦除命令中的任何一个模块失败，那么 `erase_fail` 中断被发出。失败模块的地址在 `status` 组的 `err_block_addr0` 寄存器中被更新。

锁定操作

NAND 闪存控制器支持以下功能：

- 闪存锁定 — NAND 闪存控制器支持所有闪存锁定操作。
闪存器件本身可能对这些功能的支持有限。如果器件不支持锁定功能，那么闪存控制器忽略这些命令。
- 锁紧 — 通过锁紧功能，NAND 闪存控制器可以防止锁定状态被更改。存储器被锁紧后，闪存控制器必须在任何闪存区域锁定或解锁之前被复位。

解锁一系列的存储器模块

要解锁几个存储器模块，请执行以下步骤：

1. 写入命令寄存器，将 `CMD_MAP` 域设置为 2 以及 `BLK_ADDR` 域设置为解锁区域的起始地址。
2. 将 0x10 写入 Data 寄存器。
3. 写入命令寄存器，将 `CMD_MAP` 域设置为 2 以及 `BLK_ADDR` 域设置为解锁区域的结束地址。
4. 将 0x11 写入 Data 寄存器。

当解锁一系列模块时，起始模块地址必须少于结束模块地址。否则，NAND 闪存控制器会出现不确定的行为。

锁定所有存储器模块

要锁定整个存储器：

1. 写入命令寄存器，将 `CMD_MAP` 域设置为 2 以及 `BLK_ADDR` 域设置为任何存储器地址。
2. 将 0x21 写入 Data 寄存器。

对所有存储器模块设置锁紧

应用锁紧后，解锁的区域不能被锁定，并且锁定的区域不能被解锁。

要锁紧整个存储器：

1. 写入命令寄存器，将 `CMD_MAP` 域设置为 2 以及 `BLK_ADDR` 域设置为任何存储器地址。
2. 将 0x31 写入 Data 寄存器。

要禁用锁紧，复位存储控制器。

传输模式操作

您可以在以下其中之一的数据传输模式中配置 NAND 闪存控制器：

- 默认区域传输模式
- 备用区域传输模式
- 主要 + 备用区域传输模式

NAND 闪存控制器从 config 组中 transfer_spare_reg 寄存器的设置决定默认传输模式。使用 MAP10 命令将传输模式动态地从现有模式更改为新模式。所有接下来的命令都在新模式中传输。必须注意传输模式在逻辑数据传输边界更改。例如：

- 单页面读或写情况中的页面的开始或结束。
- 完整多页面流水线读或写命令的开始或结束。

请参考第 10 - 8 页的“MAP10 命令”，以便了解关于 MAP10 命令的详细信息。
表 10 - 17 列出了 MAP10 传输模式命令的功能性，以及它们到 config 组中 transfer_spare_reg 寄存器的映射。

表 10 - 17. transfer_spare_reg 和 MAP10 传输模式命令

transfer_spare_reg	MAP10 传输模式命令	所得 NAND 闪存控制器模式
0	0x42	主要 (1)
0	0x41	备用
0	0x43	主要 + 备用
1	0x42	主要 + 备用 (1)
1	0x41	备用
1	0x43	主要 + 备用

表 10 - 17 注释：

(1) 默认访问模式 (0x42) 根据 transfer_spare_reg 的值映射到主要（仅为）或主要 + 备用模式。

对默认区域访问进行配置

如果传输模式之前被更改为备用区域或主要 + 备用区域，那么只需要对默认区域访问进行配置。要配置默认区域访问：

1. 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为任何模块。
2. 将 0x42 写入 Data 寄存器。

NAND 闪存控制器从 config 组中 transfer_spare_reg 寄存器的设置决定默认区域传输模式。如果它被设置为 1，那么传输模式变为主要 + 备用区域，否则它是主要区域。

对备用区域访问进行配置

要只对闪存器件的备用区域进行访问，请使用 MAP10 命令来设置 NAND 闪存控制器，以便只读取或写入器件的备用区域。设置闪存控制器之后，请使用 MAP01 读和写命令访问相应模块和页面地址的备用区域。要配置 NAND 闪存控制器以便只访问备用区域，请执行以下步骤：

1. 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为目标模块。
2. 将 0x41 写入 Data 寄存器。

对主要 + 备用区域访问进行配置

要配置 NAND 闪存控制器以访问主要 + 备用区域：

1. 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为目标模块。
2. 将 0x43 写入 Data 寄存器。

Read-Modify-Write 操作

要读取指定的页面或修改页面中一些字、字节或位，请使用 RMW 操作。一个读命令将所需数据从闪存存储器复制到页面缓冲器。从而可以使用 MAP00 缓冲器读和写命令修改缓冲器中的信息并且发出另一个命令以便将该信息写回存储器。

read-modify-write 命令在整个页面上操作。该命令也用于复制类型操作，其中页面的大部分被保存到一个新位置。在该操作类型中，NAND 闪存控制器读取数据，修改页面中指定的字的数量，然后将修改的页面写入新的位置。



因为数据在闪存器件的页面缓冲器中被修改，那么 NAND 闪存控制器 ECC 硬件不在 RMW 操作中使用。软件必须在 RMW 操作期间更新 ECC。



对于硬件 ECC 的 read-modify-write 命令，整个页面必须被读入系统存储器、被修改，然后被写回到闪存而不依赖 RMW 功能。

Read-Modify-Write 操作流程

1. 流程通过从存储器读取页面开始：

- 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为所需模块的起始地址。
- 将 0x60 写入 Data 寄存器。

该步骤使闪存器件中页面缓冲器的页面可用。

2. 该步骤提供目的页面地址：

- 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为所需模块的目的地址。
- 将 0x61 写入 Data 寄存器。

该步骤启动页面编程并且对器件提供目的地址。

3. 使用 MAP00 页面缓冲器读和写命令来修改页面缓冲器中的数据。

4. 将页面缓冲器数据写回到存储器：

- 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为相同目的地址。
- 将 0x62 写入 Data 寄存器。

该步骤执行写操作。

器件完成加载操作后，NAND 闪存控制器发出一个 load_comp 中断。当主机发出写命令并且器件完成编程操作时，一个 program_comp 中断被发出。

如果页面编程操作（作为 RMW 操作的一部分）导致了器件中编程失败，那么 program_fail 中断被发出。失败页面的模块和页面地址在 status 组的 err_block_addr0 和 err_page_addr0 寄存器中被更新。

Copy-back 操作

NAND 闪存控制器支持 copy back 操作。然而，闪存器件可能对该功能的支持有限。如果您尝试在不支持 copy-back 的器件上执行 copy-back 操作，那么 NAND 闪存控制器触发一个中断。如果源模块在目标模块被指定之前没有被指定，或如果目标模块在遵循源模块规范的下一个命令中没有被指定，那么中断也被触发。

NAND 闪存控制器在 copy-back 命令的情况中不可以进行 ECC 验证。闪存控制器复制 ECC 数据，但是在复制操作期间不检查它。Altera 建议如果闪存控制器中实现的 ECC 足够强，那么仅使用 copy-back，以便下一个访问可以纠正累加错误。

8-bit 值 $\langle PP \rangle$ 指定 copy-back 的页面数。通过该功能，NAND 闪存控制器可以使用单命令复制多个连续的页面。当发出 copy-back 命令时，闪存控制器在后台执行操作。闪存控制器暂停其它的命令直到目前 copy-back 完成。

对于一个多平面器件，如果 config 组的 multiplane_operation 寄存器中的 flag 位被设置为 1，那么多平面 copy-back 可作为一个选项。在该情况中，指定的模块地址必须是平面对齐的并且 $\langle PP \rangle$ 值必须指定复制的页面总数为平面数的倍数。递增页面地址之前，对于器件中整个平面数，模块地址继续递增，保持页面地址固定。

当闪存控制器已经完成所有 $\langle PP \rangle$ 页面的 copy-back 操作时，一个 pipe_cpyback_cmd_comp 中断被生成。如果任何页面编程操作（作为 copy back 操作的一部分）在器件中导致编程失败，那么 program_fail 中断被发出。失败页面的模块和页面地址在 status 组中的 err_block_addr0 和 err_page_addr0 寄存器中更新。

复制一个存储区域（单平面）

要从一个存储位置到另一个存储位置复制 $\langle PP \rangle$ 页面：

1. 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为复制的区域的起始地址。
2. 将 0x1000 写入 Data 寄存器。
3. 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为写入的新区域的起始地址。
4. 将 0x11 $\langle PP \rangle$ 写入 Data 寄存器，其中 $\langle PP \rangle$ 是复制的页面数。

复制一个存储区域（多平面）

要从一个存储位置到另一个存储位置复制 $\langle PP \rangle$ 页面：

1. 将 config 组的 multiplane_operation 寄存器中的 flag 位设置为 1。
2. 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为复制区域的起始位置。地址必须是平面对齐的。
3. 将 0x1000 写入 Data 寄存器。
4. 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为写入的新区域的起始地址。该地址必须也是平面对齐的。
5. 将 0x11 $\langle PP \rangle$ 写入 Data 寄存器，其中 $\langle PP \rangle$ 是复制的页面数。

参数 $\langle PP \rangle$ 必须是器件中平面数的倍数。

流水线预读和预写操作

NAND 闪存控制器支持流水线预读和预写操作。然而，闪存器件可能对该功能的支持有限。如果器件不支持流水线预读或预写，那么闪存控制器将这些命令处理为标准读或写。

NAND 闪存控制器可以最多处理 4 个未决的流水线命令，以闪存控制器接收到命令的顺序排列。闪存控制器操作队列开头的流水线命令直到与流水线命令相一致的所有页面被执行。然后闪存控制器弹出队列开头的流水线命令并且继续执行队列中的下一个流水线命令。

流水线预读功能支持闪存存储器的连续读取。接收到一个流水线读命令时，闪存控制器立即对器件发出一个加载命令。当数据使用 MAP01 命令以连续的或多平面地址码型被读取时，闪存控制器对来自闪存器件的连续的数据流保持额外的缓存或多平面读命令序列。

流水线预写功能支持闪存存储器的连续的写入。当数据使用 MAP01 命令以连续的或多平面地址码型被写入时，NAND 闪存控制器对到闪存器件的连续的数据流保持缓存或多平面命令序列。

MAP01 命令必须以流水线命令发送到 NAND 闪存控制器的相同序列读或写页面。如果主机发出多个流水线命令，那么页面必须以流水线命令被发出的顺序被读取或写入。在完成第一个流水线命令之前不可能对第二个流水线命令读或写页面。如果流水线序列由 MAP01 命令损坏，那么 pipe_cmd_err 中断被发出，并且闪存控制器清除流水线命令序列。闪存控制器使用正常的页面读或写序列执行违规输入 MAP01 读或写请求。

对于一个支持多平面编程的多平面器件，必须将 config 组的 multiplane_operation 寄存器中的 flag 位设置为 1。在该情况中，数据被交错到连续的模块的页面容量块中。

流水线预读命令可以使用这种交错方式从队列读取数据。参数 *<PP>* 表示页面的总数为平面数的倍数，并且模块地址必须是平面对齐的，从而对于每个页面容量的数据块在递增模块地址的同时保持页面地址一致。从每个平面读取后，NAND 闪存控制器递增页面地址并且将模块地址复位到初始地址。也可以在多平面模式中使用流水线预写命令。写操作和读操作进行的工作相似，可以在递增模块地址的同时保持页面地址一致直到所有平面被写。



相同的 4 入口队列用于对流水线预读和预写命令排列地址和页面数。这一通用性要求您使用 MAP01 命令在处理下一个流水线命令之前读取流水线预读命令的所有页面。同样，必须在处理下一个流水线命令之前写入与流水线预写命令相关的所有页面。

因为 config 组中 multiplane_operation 寄存器的 flag 位的值决定流水线预读或预写行为，所以它仅可以在流水线寄存器为空时被更改。

当主机发出一个流水线预读命令，并且闪存控制器闲置时，加载操作会立即发生。



预读命令不对主机返回数据，并且预写命令不对闪存地址写数据。NAND 闪存控制器加载读数据。只有当主机发出 MAP01 命令读数据时，读数据才被返回到主机。同样，闪存控制器加载写数据，并且只有当主机发出 MAP01 命令写数据时才将写数据写入闪存。

当 NAND 闪存控制器已经完成处理流水线命令并且已经从队列中丢弃该命令时，pipe_cpyback_cmd_comp 中断被生成。这时，主机可以发送另一个流水线命令。一个流水线命令从队列弹出，并且当闪存控制器已经开始处理流水线命令的最后页面时一个中断被发出。因此，在流水线读命令的情况中，最后页面加载之前，pipe_cpyback_cmd_comp 中断被发出，以及在流水线写命令的情况中，开始编程最后页面的数据传输时，pipe_cpyback_cmd_comp 中断被发出。

当最后页面编程操作在流水线写命令情况中完成时，额外的 program_comp 中断被生成。

如果器件命令设置要求 NAND 闪存控制器对流水线读命令中的最后页面发出一个加载命令，那么 load_comp 中断在最后页面加载操作完成后被生成。

对于流水线写命令，如果任何页面编程导致器件中的失败，那么 program_fail 中断被发出。失败页面的模块和页面地址在 status 组的 err_block_addr0 和 err_page_addr0 寄存器中更新。

流水线命令排序（例如缓存和多平面）器件中的高级命令。当 NAND 闪存控制器接收到多页面读或写流水线命令时，它根据以下 config 组中寄存器的设置排序发送到器件的命令：

- cache_read_enable
- cache_write_enable
- multiplane_operation

对于一个支持缓存读序列的器件，cache_read_enable 寄存器的 flag 位必须设置为 1。NAND 闪存控制器将每个多页面流水线读命令排序为一个缓存读序列。对于一个支持缓存编程命令序列的器件，cache_write_enable 必须被设置。闪存控制器将每个多页面写流水线命令排序为一个缓存写序列。

对于具有多平面并且支持多平面编程命令的器件，config 组中的 NAND 闪存控制器寄存器 multiplane_operation 必须被设置。接收到多页面流水线写命令时，闪存控制器使用多平面编程命令排序器件并且期待主机以奇偶模块递增地址模式对闪存控制器传输数据。

对流水线预读设置一个单区域

要对流水线预读设置一个区域，请执行以下步骤：

1. 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为预读模块的起始地址。
2. 将 0x20<PP> 写入 Data 寄存器，其中值 0 设置该命令为预读并且 <PP> 是预读的页面数。页面一定不可以跨越模块边界。如果模块边界被跨越，那么 NAND 闪存控制器生成一个不被支持的命令 (unsup_cmd) 中断并且丢弃命令。

预读命令对闪存器件进行提示，以便之前的页面缓冲器操作一完成，它就开始加载页面缓冲器中的下一个页面。设置预读之后，使用 MAP01 命令准确地读取数据。在 MAP01 命令中，指定和预读中相同的起始地址。

如果接收到的跟随流水线预读请求的读命令不到预读的页面，那么中断位被设置为 1 并且流水线预读或预写寄存器被清零。您必须发出一个新流水线预读请求以便重新加载相同的数据。您必须使用 MAP01 命令在 NAND 闪存控制器返回闲置状态之前读取预读的所有数据。

对流水线预写设置一个单区域

要对流水线预写设置一个区域：

1. 写入命令寄存器，将 CMD_MAP 域设置为 2 以及 BLK_ADDR 域设置为预写模块的起始地址。
2. 将 0x21<PP> 写入 Data 寄存器，其中值 1 将该命令设置为预写并且 <PP> 是预写的页面数。页面一定不可以跨越模块边界。如果模块边界被跨越，那么 NAND 闪存控制器生成一个不被支持的命令 (unsup_cmd) 中断并且丢弃命令。

设置预写之后，使用 MAP01 命令准确地写数据。在 MAP01 命令中，指定和预写中相同的起始地址。

如果接收到的跟随流水线预写请求的写命令不到预写的页面，那么中断位被设置为 1 并且流水线预读或预写寄存器被清零。您必须发出一个新流水线预写请求以配置写逻辑。

您必须使用 MAP01 命令在 NAND 闪存控制器返回闲置状态之前写入预写的所有数据。


NAND 闪存控制器地址映射和寄存器定义

 地址映射和寄存器定义位于该手册卷附带的 [hps.html](#) 文件中。点击链接以打开文件。

要查看模块说明和基地址，找到并且点击以下链接以查看模块实例：

- [nanddata](#)
- [nandregs](#)

要查看寄存器和域说明，找到并且点击寄存器名称。寄存器地址是相对于每个模块实例的基地址的偏移。

 所有模块的基地址也在 *Cyclone V 器件手册* 第 3 卷的 *Introduction to the Hard Processor System* 章节中列出。

文档修订历史

表 10 - 18 列出了该文档的修订历史。

表 10 - 18. 文档修订历史

日期	版本	修订内容
2012 年 11 月	1.2	<ul style="list-style-type: none">■ 支持一个 8-bit 器件■ 显示了额外所支持的模块容量■ 坏模块标识处理
2012 年 5 月	1.1	添加了编程模型部分
2012 年 1 月	1.0	首次发布。

