

(1) 所有数据转移。

(2) 通过软件禁止该通道。

注：必须执行内存到内存的传输与低通道优先，否则，其它 DMA 通道不能进入总线，直到内存到内存的传输已经完成，或其他 AHB 的控制无法执行任何处理。

内存到内存的处理，如图 11-4 所示。



图 11-4 在 DMA 控制器的流控制下内存到内存的处理

### 7. 在 DMA 控制器的流控制下外设到外设的处理

当处理的不是脉冲大小的倍数时，用下面的信号：

(1) 单一和脉冲请求 (DMACBREQ and DMACSREQ) 来源是外设信号。

(2) 脉冲请求 (DMACBREQ) 目标是外设信号。

外设到外设的处理，如图 11-5 所示。

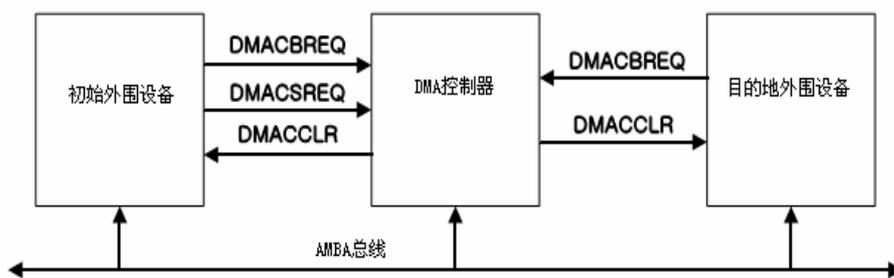


图 11-5 由脉冲和单一请求组成的外设到外设的处理

源外设遵循同样程序，当作描述外设到内存的 DMA 控制器的流量控制。目标外设遵循同样的程序，当作描述内存到外设的处理下外设的流量控制。下一个 LLI 装载时，所有读与写的传输是完整的。可以使用 DMACTC 的信号表明，过去的的数据已传输到外设上。如表 11-2 所示。

表 11-2 DMA 控制器的流量控制

传输方向	请求发生器	请求信号使用
外设到内存	外设	DMACBREQ
内存到外设	外设	DMACBREQ
内存到内存	DMA 控制器	None
外设到外设	外设	Src: DMACBREQ

## 11.4 信号时序

DMA 信号的时序行为描述如下：

(1) DMA 请求信号 DMAC{L}(B/S)REQ<sub>x</sub>

通知 DMA 控制器，该外设准备按指定的大小进行 DMA 传输。

高有效位。由 DMA 控制器取样，关于 HCLK 的实际优势。DMA 请求信号用于连接 DMACCLR 信号来实现握手。

(2) DMA 的承认或明确的 DMACCLR<sub>x</sub>

简单说明一个 DMA 传输的完成。高有效位。

(3) DMA 的终端计数的 DMACTC<sub>x</sub>

简单说明数据包的最后已经准备。高有效位。

注：如果 DMA 请求来源不使用相同的时钟作为 DMA 控制器，则在 DMACSync 寄存器中，必须通过设置相关的位请求同步。

## 11.5 功能时序图

外设表明，一个 DMA 请求保持有效状态。该 DMACCLR 信号声称，当结束数据项目已被传输时，DMA 控制器表明 DMACCLK 信号，如图 11-6 所示。

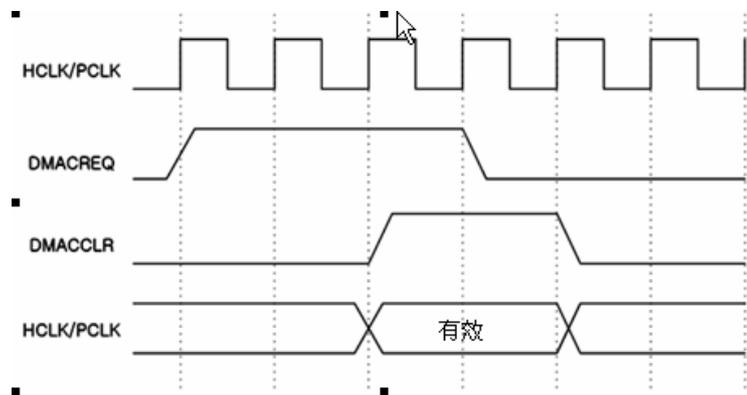


图 11-6 DMA 接口时序

## 11.6 程序员的模式

### 1. 设计 DMA 控制器

所有 AHB 从执行总线的处理必须是 32 位带宽。当编程 DMA 控制器时，消除尾端的问题。

### 2. 启动 DMA 控制器

在 DMA 配置寄存器，启动 DMA 控制器设置 DMA 的使能位。

### 3. 禁用 DMA 控制器

禁用 DMA 控制器采取下列步骤：

(1) 读取 DMACEnbldChns 寄存器，并确保所有 DMA 通道已被禁用。如果任何通道是有效的，提交禁用 DMA 通道。

(2) 通过在 DMA 配置寄存器写入 DMA 有效位，停用 DMA 控制器。

### 4. 启动 DMA 通道

在启动 DMA 通道设置，控制有关 DMA 通道配置寄存器。

注：通道必须在初始化之前，它是启用的。此外，DMA 控制器必须设置之前，启用通道。

### 5. 禁用 DMA 通道

DMA 通道可以被禁用是在以下三个方面：

- (1) 直接写入通道使能位。如果使用这个方法任何在 FIFO 中的突出的数据将丢失。
- (2) 使用有效和停止位连接通道使能位。
- (3) 直到传输完成。该通道是自动禁止。

禁用 DMA 通道，并丢失 FIFO 中数据：

在有关通道配置寄存器中，清楚有关通道使能位。当前的 AHB 传输完成和通道被禁用。在 FIFO 中，任何数据失去。

禁止 DMA 通道，并不失去的数据。

禁止 DMA 通道并不会丢失 FIFO 中数据的步骤如下：

(1) 在有关的通道配置寄存器中设置停止位。忽略进一步的 DMA 请求。

(2) 在有关的通道配置寄存器轮询有效位，直到它满足条件。该位指示是否通道中还有没传输的数据。

(3) 在有关通道配置寄存器中，清除有关通道使能位。

## 6. 建立一个新的 DMA 传输

建立一个新的 DMA 传输的步骤：

(1) 如果通道没有设置用于 DMA 处理的边：

- a. 读取 DMACEnbldChns 控制器寄存器，并找出哪些通道是无效的。
- b. 选择一个无效的通道，它有请求优先级。

(2) 执行 DMA 控制器。

## 7. 停止 DMA 通道

在有关的 DMA 通道配置寄存器中，设置停止位。用于当前源请求，直到停止位被清除，任何另外的来源的 DMA 请求被忽略。

## 8. 编程 DMA 通道

编程 DMA 通道的步骤：

(1) 决定是否使用安全 DMAC (SDMAC) 或一般 DMAC (DMAC)。在正常状态下使用一般 DMAC，禁用系统控制器的安全 DMA 控制寄存器 (sdma\_sel)。(重置 SDMAC)

(2) 根据优先需要，选择自由 DMA 通道。DMA 通道 0 有最高优先权，DMA 通道 7 优先权最低。

(3) 通过写 DMACIntTCClr 和 DMACIntErrClr 寄存器，清除通道中要用到的未处理的中断。先前的通道操作可能使剩余的中断有效。

(4) 写源地址到 DMACCxSrcAddr 寄存器中。

(5) 写目标地址到 DMACCxSrcAddr 寄存器中。

(6) 将下一个 LLI 的地址写入 DMACCxLLI 寄存器。如果是传输单一的包，那么必须写入

该寄存器。如表 11-3 所示。

表 11-3 写下一个 LLI 地址到 DMACCxLLI 寄存器

偏移	内容
下一个 LLI 地址	用于下一个传输的源地址
下一个 LLI 地址 0x04	用于下一个传输的目标址
下一个 LLI 地址 0x08	用于下一个传输的下一个 LLI 地址
下一个 LLI 地址 0x0C	用于下一个传输的 DMACCxControl0 数据
下一个 LLI 地址 0x10	下一个传输的 DMACCxControl1 数据

(7) 写控制信息在 DMACCxControl 寄存器中。

(8) 写通道配置信息到 DMACCxConfiguration 寄存器中。如果使能位被设置，那么该 DMA 通道自动启用。

## 11.7 寄存器描述

有四个 DMA 控制器名称，其为 DMAC0, DMAC1, SDMAC0 和 SDMAC1。

DMAC0, DMAC1, SDMAC0 和 SDMAC1 寄存器基地址分别是 0x7500\_0000, 0x7510\_0000, 0x7DB0\_0000 和 0x7DC0\_0000。

用于 OneNAND 控制器的页面访问功能是通过 DMAC0 和 SDMAC0 的通道 3 添加的。

DMA 寄存器的位置。如表 11-4 所示，对 DMA 寄存器进行概括的描述。

表 11-4 DMA 寄存器概要

名称	类型	宽度	描述	偏移量	复位值
DMACIntStatus	读	8	该寄存器提供 DMA 控制器的中断状态。高位指示一个特殊的 DMA 通道中断有效。	0x000	0x00
DMACIntTCStatus	读	8	该寄存器是用来判断处理过程中是否产生中断，高位指出传输被完成。	0x004	0x00
DMACIntTCClear	写	8	当写入该寄存器，每个数据位都是高位将使 DMACIntTCStatus 和 DMACRawIntTCStatus 寄存器清除。数	0x008	-

			据位是低位则对相应的寄存器没有影响。		
DMACIntErrorStatus	读	8	这寄存器是用来判断由于一个错误是否有中断产生。	0x00C	0x00
DMACIntErrClr	写	8	当写这个寄存器时，每个数据位都是高位将使 DMACIntErrorStatus 和 DMACRawIntErrorStatus 寄存器清除，数据位是低位则对相应的寄存器没有影响。	0x010	-
DMACRawIntTCStatus	读	8	屏蔽前，该寄存器提供 DMA 终端计数中断的原始状态。高位指出中断请求优先屏蔽，被激活。	0x014	-
DMACRawIntErrorStatus	读	8	屏蔽前，该寄存器提供 DMA 错误的原始状态。高位指出中断请求优先屏蔽，被激活。	0x018	-
DMACEnbldChns	读	8	寄存器显示 DMA 通道被激活，高位指出中断请求优先屏蔽，被激活。	0x01C	0x00
DMACSoftBReq	读/写	16	该寄存器通过软件允许 DMA 的脉冲模块产生。	0x020	0x0000
DMACSoftSReq	读/写	16	这寄存器通过软件允许 DMA 的单一请求产生。	0x024	0x0000
DMACSoftLReq	读/写	16	该寄存器通过软件允许 DMA 的最后脉冲模块产生。	0x028	0x0000
DMACSoftLSReq	读/写	16	该寄存器通过软件允许 DMA 的最后单一请求产生。	0x02C	0x0000
DMACConfiguration	读/写	3	该寄存器是用来配置 DMA 控制器。	0x030	0b000
DMACSync	读/写	16	该寄存器启用或禁止用于 DMA 请求信号的同步逻辑。	0x034	0x0000
DMAC0SrcAddr	读/写	32	DMA 通道 0 的初始化地址。	0x100	0x00000000

DMACC0DestAddr	读/写	32	DMA 通道 0 的目标地址。	0x104	0x00000000
DMACCOLLI	读/写	32	DMA 通道 0 的链表地址。	0x108	0x00000000
DMACC0Control0	读/写	32	DMA 通道 0 控制器 0。	0x10C	0x00000000
DMACC0Control1	读/写	32	DMA 通道 0 控制器 1。	0x110	0x00000000
DMACC0Configuration	读/写	19	DMA 通道 0 配置寄存器。	0x114	0x000000
DMACC1SrcAddr	读/写	32	DMA 通道 1 的初始化地址。	0x120	0x00000000
DMACC1DestAddr	读/写	32	DMA 通道 1 的目标地址。	0x124	0x00000000
DMACC1LLI	读/写	32	DMA 通道 1 的链表地址。	0x128	0x00000000
DMACC1Control0	读/写	32	DMA 通道 1 控制器 0。	0x12C	0x00000000
DMACC1Control1	读/写	32	DMA 通道 1 控制器 1。	0x130	0x00000000
DMACC1Configuration	读/写	19	DMA 通道 1 配置寄存器。	0x134	0x000000
DMACC2SrcAddr	读/写	32	DMA 通道 2 的初始化地址。	0x140	0x00000000
DMACC2DestAddr	读/写	32	DMA 通道 2 的目标地址。	0x144	0x00000000
DMACC2LLI	读/写	32	DMA 通道 2 的链表地址。	0x148	0x00000000
DMACC2Control0	读/写	32	DMA 通道 2 控制器 0。	0x14C	0x00000000
DMACC2Control1	读/写	32	DMA 通道 2 控制器 1。	0x150	0x00000000
DMACC2Configuration	读/写	19	DMA 通道 2 配置寄存器。	0x154	0x000000
DMACC3SrcAddr	读/写	32	DMA 通道 3 的初始化地址。	0x160	0x00000000
DMACC3DestAddr	读/写	32	DMA 通道 3 的目标地址。	0x164	0x00000000
DMACC3LLI	读/写	32	DMA 通道 3 的链表地址。	0x168	0x00000000
DMACC3Control0	读/写	32	DMA 通道 3 控制器 0。	0x16C	0x00000000
DMACC3Control1	读/写	32	DMA 通道 3 控制器 1。	0x170	0x00000000
DMACC3Configuration	读/写	19	DMA 通道 3 配置寄存器。	0x174	0x000000
DMACC4SrcAddr	读/写	32	DMA 通道 4 的初始化地址。	0x180	0x00000000
DMACC4DestAddr	读/写	32	DMA 通道 4 的目标地址。	0x184	0x00000000
DMACC4LLI	读/写	32	DMA 通道 4 的链表地址。	0x188	0x00000000
DMACC4Control0	读/写	32	DMA 通道 4 控制器 0。	0x18C	0x00000000
DMACC4Control1	读/写	32	DMA 通道 4 控制器 1。	0x190	0x00000000

DMACC4Configuration	读/写	19	DMA 通道 4 配置寄存器。	0x194	0x00000
DMACC5SrcAddr	读/写	32	DMA 通道 5 的初始化地址。	0x1A0	0x00000000
DMACC5DestAddr	读/写	32	DMA 通道 5 的目标地址。	0x1A4	0x00000000
DMACC5LLI	读/写	32	DMA 通道 5 的链表地址。	0x1A8	0x00000000
DMACC5Control0	读/写	32	DMA 通道 5 控制器 0。	0x1AC	0x00000000
DMACC5Control1	读/写	32	DMA 通道 5 控制器 1。	0x1B0	0x00000000
DMACC5Configuration	读/写	19	DMA 通道 5 配置寄存器。	0x1B4	0x00000
DMACC6SrcAddr	读/写	32	DMA 通道 6 的初始化地址。	0x1C0	0x00000000
DMACC6DestAddr	读/写	32	DMA 通道 6 的目标地址。	0x1C4	0x00000000
DMACC6LLI	读/写	32	DMA 通道 6 的链表地址。	0x1C8	0x00000000
DMACC6Control0	读/写	32	DMA 通道 6 控制器 0。	0x1CC	0x00000000
DMACC6Control1	读/写	32	DMA 通道 6 控制器 1。	0x1D0	0x00000000
DMACC6Configuration	读/写	19	DMA 通道 6 配置寄存器。	0x1D4	0x00000
DMACC7SrcAddr	读/写	32	DMA 通道 7 的初始化地址。	0x1E0	0x00000000
DMACC7DestAddr	读/写	32	DMA 通道 7 的目标地址。	0x1E4	0x00000000
DMACC7LLI	读/写	32	DMA 通道 7 的链表地址。	0x1E8	0x00000000
DMACC7Control0	读/写	32	DMA 通道 7 控制器 0。	0x1EC	0x00000000
DMACC7Control1	读/写	32	DMA 通道 7 控制器 1。	0x1F0	0x00000000
DMACC7Configuration	读/写	19	DMA 通道 7 配置寄存器。	0x1F4	0x00000

### 11.7.1. 中断状态寄存器 DMACIntStatus

该 DMACIntStatus 寄存器是只读类型，并且在屏蔽后指示中断状态。高位指示一个特殊的 DMA 通道中断请求有效。由于错误或者终端计数中断请求，该请求产生。

显示 DMACIntStatus 寄存器的位分配，如表 11-5 所示。

表 11-5 DMACIntStatus 寄存器的位分配

DMACIntStatus	位	类型	功能
---------------	---	----	----

IntStatus	[7:0]	R	DMA 的地位屏蔽后中断
-----------	-------	---	--------------

### 11.7.2. 中断终端计数状态寄存器，DMACIntTCStatus

DMACIntTCStatus 寄存器是只读类型，并且屏蔽后显示终端计数状态。如果结合中断请求，这个寄存器必须用于关联 DMACIntStatus 寄存器。DMACINTCOMBINE，用于中断请求。

如果使用 DMACINTTC 的中断请求，那么就得读 DMACIntTCStatus 寄存器来确定中断请求的来源。如表 11-6 所示。

表 11-6 显示 DMACIntTCStatus 寄存器的位分配

DMACIntTCStatus	位	类型	功能
IntTCStatus	[7:0]	读	中断终端计数状态

### 11.7.3. 中断终端计数清除寄存器，DMACIntTCClear

DMACIntTCClear 寄存器是只写类型，用于清除中断计数中断请求。

当写入这个寄存器，每个数据位设置为高位，原因是在状态寄存器中的相应位被清除。数据位为低位，不影响该寄存器中的相应位。如表 11-7 所示，显示了 DMACIntTCClear 寄存器的位分配。

表 11-7 DMACIntTCClear 寄存器的位分配

DMACIntTCClear	位	类型	功能
IntTCClear	[7:0]	写	终端计数请求清除

### 11.7.4. 中断错误状态寄存器，DMACIntErrorStatus

DMACIntErrorStatus 寄存器是只读类型，屏蔽后显示错误请求的状态。如果结合中断请求，该寄存器必须被用于关联 DMACIntStatus 寄存器，DMACINTCOMBINE 用于请求中断。

如果 DMACINTERROR 中断请求只用于 DMACIntErrorStatus 寄存器，需要读取。表 11-8 所示，显示 DMACIntErrorStatus 寄存器的位分配。

表 11-8 DMACIntErrorStatus 寄存器的位分配

DMACIntErrorStatus	位	类型	功能
IntErrorStatus	[7:0]	读	中断错误状态

### 11.7.5. 中断错误清除寄存器，DMACIntErrClr

DMACIntErrClr 寄存器是只写类型，它用于清除错误中断请求。当读这个寄存器时，每个数据为是高位，原因是在状态寄存器中的相应位被清除。当数据位是低位，不影响该寄存器中的相应位。如表 11-9 所示，显示 DMACIntErrClr 寄存器的位分配。

表 11-9 DMACIntErrClr 寄存器的位分配

DMACIntErrClr	位	类型	功能
IntErrClr	[7:0]	写	中断错误清除

### 11.7.6. 原始中断终端计数状态寄存器，DMACRawIntTCStatus

DMACRawIntTCStatus 寄存器是只读类型。它指示其中的 DMA 通道请求一个传输完成（终端计数中断）。高位指示，终端计数中断请求是有效的，优先于屏蔽。如表 11-10 所示，显示 DMACRawIntTCStatus 寄存器的位分配。

表 11-10 DMACRawIntTCStatus 寄存器的位分配

DMACRawIntTCStatus	位	类型	功能
RawIntTCStatus	[7:0]	读	终端计数中断状态优先于屏蔽

### 11.7.7. 原始错误中断状态寄存器，DMACRawIntErrorStatus

DMACRawIntErrorStatus 寄存器是只读类型，它指示其中 DMA 通道屏蔽前，请求传输完成。高位指示终端的计数中断请求优先于屏蔽被激活。如表 11-11 所示，显示 DMACRawIntErrorStatus 寄存器的位分配。

表 11-11 DMACRawIntErrorStatus 寄存器的位分配

DMACRawIntErrorStatus	位	类型	功能
-----------------------	---	----	----