AN14300

MCX Nx4x：释放eDMA控制器的能力

第1.0版—2024年9月23日 应用笔记

文档信息

信息	内容
关键词	AN14300、eDMA、总线主设备、DMA、MCX Nx4x
摘要	本应用笔记通过涵盖以下主题提供了实用的知识：eDMA控制器的介绍和概述，MCX Nx4x eDMA模块的功能，eDMA和DMA多路复用器（DMAMUX）之间的交互，以及针对应用程序的配置建议和示例。

1 介绍

MCX Nx4x系列微控制器将Arm Cortex-M33 TrustZone核与CoolFlux BSP32、PowerQuad DSP协处理器以及多个以150 MHz运行的高速连接选项相结合。它具有卓越的处理能力和先进的集成度，非常适合从电机控制和工业自动化到音频处理和通信系统等各种高要求应用。然而，这些应用要实现最佳性能，高效的数据管理至关重要。

增强型直接内存存取（eDMA）实现了存储器和外设之间的高效数据传输，可减轻CPU的工作负载并提升系统性能。MCX Nx4x系列微控制器提供了一个多功能的eDMA控制器，该控制器可配置以满足广泛的数据传输需求。本应用笔记通过涵盖以下主题提供了实用的知识：eDMA控制器的介绍和概述，MCX Nx4x eDMA模块的功能，eDMA和DMA多路复用器（DMAMUX）之间的交互，以及针对应用程序的配置建议。本文使用示例来演示越来越复杂的eDMA配置。

1.1 eDMA控制器概述

eDMA控制器支持在无CPU干预的情况下，将数据从一个存储映射位置移动到另一个存储映射位置。一旦配置并启动，eDMA控制器将与中央处理单元（CPU）并行运行，执行原本由CPU处理的数据传输。这样就减轻了CPU的负载，并相应地提高了系统性能。图1所示为DMA控制器提供的功能。

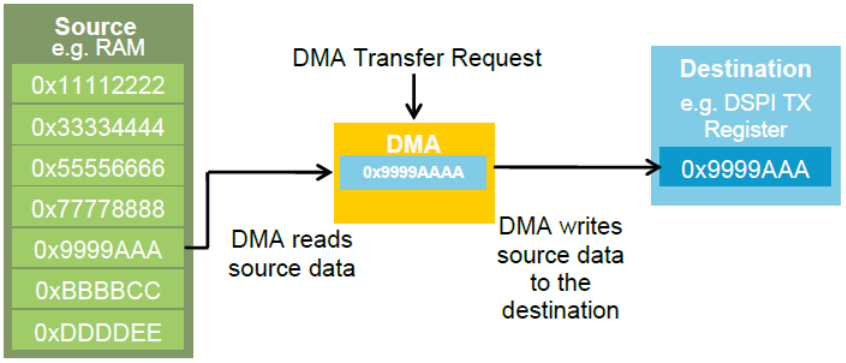

图1. eDMA运行概览

1.2 MCX Nx4x eDMA模块描述

eDMA模块被划分为两个主要模块（图2）：

1. eDMA引擎
2. 包含每个通道的传输控制描述符（TCD）的本地存储器。

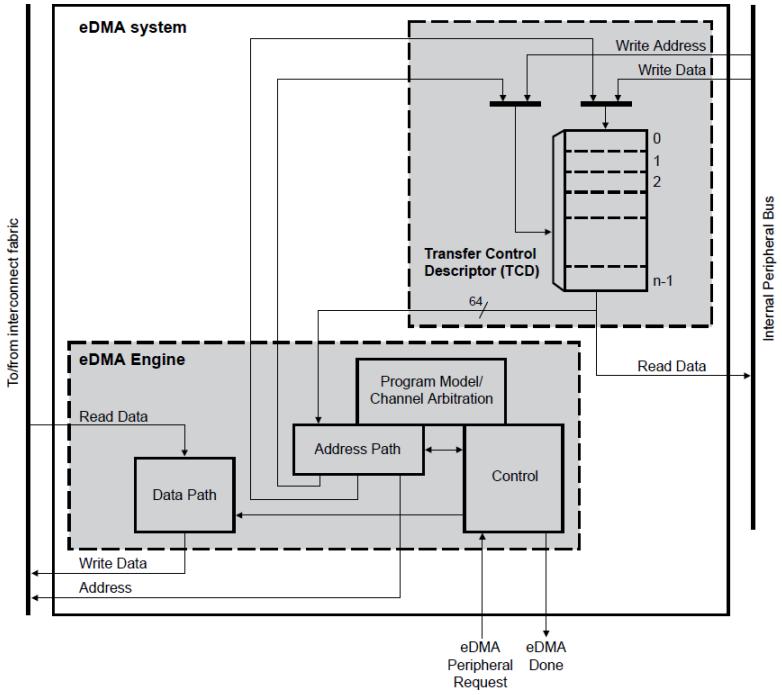

图2. eDMA的结构框图

进一步划分的子模块的详细信息如下。

1.2.1 地址路径

- 实现主通道和次通道。
- 负责为每次传输生成源和目的存储器地址。
- 允许与一个通道相关联的数据传输被更高优先级的通道抢占。
- 支持多种寻址模式，如递增、递减、固定和分散-聚集。
- 允许配置传输大小和对齐方式，以获得最佳性能。

1.2.2 数据路径

- 实现总线主设备读/写数据路径。
- 内部读、写数据总线是主输入和输出。
- 集成了硬件流控，以防止数据溢出或下溢。

1.2.3 程序/通道仲裁

- 对多个通道同时访问eDMA引擎进行管理。
- 使用优先级级别来确保关键数据首先得到处理。
- 包括轮询调度，以实现资源的公平分配。

1.2.4 控制

- 协调整个eDMA控制功能，包括启动传输、管理中断和处理错误。
- 提供各种控制寄存器，用于配置和监控。

1.2.5 TCD

- 每个eDMA通道都有一个专用的32字节传输控制描述符（TCD）模块。
- TCD进一步划分为存储控制器和存储阵列：
  - 存储控制器管理来自eDMA引擎和内部外设总线的访问。
  - 存储阵列存储每个通道的传输配置，如源地址、目的地址、偏移量和属性。这些细节在通道的TCD寄存器中指定。

1.3 MCX Nx4x eDMA控制器的特性

MCX Nx4x器件配备有两个16通道的DMA控制器。每个通道都可以独立配置要执行的传输序列的详细信息。
MCX Nx4x系列的DMAMUX允许将最多128个DMA请求信号（保留6个未使用的信号）映射到每个通道。
eDMA传输可以通过以下三种方式激活：

1. 由外设发起的硬件请求。
2. 软件启动。
3. 通道到通道的链接——传输完成后，一个通道激活另一个通道。

每个通道都可以生成中断，以指示它已部分完成或全部完成传输。还可以生成中断来指示发生了传输错误。

16个通道中的每一个都支持分散/聚集（Scatter/Gather）处理。此功能允许通道在执行完其当前配置的传输时自行加载一个新的TCD。通过动态定义和使用传输序列，此功能可以实现超过16个的扩展传输序列。

1.4 eDMA架构集成

为了使eDMA、CPU和其它主设备能够同时运行，实现了多主设备总线架构。MCX Nx4x芯片具有多个总线主设备，例如内核、快速以太网控制器和LFAST。

交叉开关（XBAR）构成了这种多主设备架构的核心。它将每个主设备连接到所需的从设备。MCX Nx4x系列中的许多器件都配备有双交叉开关架构。在这种情况下，如果一个主设备需要访问与其不在同一个交叉开关上的从设备，数据就会从一个交叉开关传递到下一个交叉开关。如果两个或多个主设备尝试联合访问同一台从设备，则会启动仲裁方案，从而消除总线竞争的风险。固定优先级和轮询仲裁方案都可用。

交叉开关以及总线主设备与从设备之间的交互如图3中的简化版本所示。在此示例中，eDMA控制器正在访问IP总线上的某个外设，与此同时CPU正在访问SRAM存储器。交叉开关已经为这种情况建立了适当的连接。

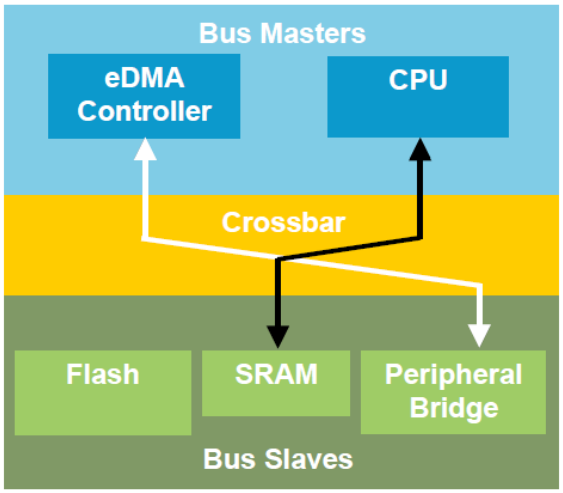

图3. 多主设备总线架构

2 eDMA数据流

基本的数据传输流程可以分为三个部分，具体解释如下：

- **发起传输**
  - **请求服务**
    如图4所示，所选通道通过发出eDMA外设请求信号来请求服务。此触发类似于软件发起的流程，其中使用了`TCDn_CSR[START]`字段。
  - **路由请求**
    内部eDMA引擎收到请求，并通过控制模块、程序模型和通道仲裁单元对其进行路由。
  - **通道仲裁**
    在下一个周期中，固定优先级算法（可选与轮询相结合）选择激活的通道。
  - **访问TCD描述符**
    然后，使用所选的通道号从64位宽的TCD存储器中访问并读取相应的TCD描述符。此描述符被加载到地址路径的主通道或次通道执行寄存器中。

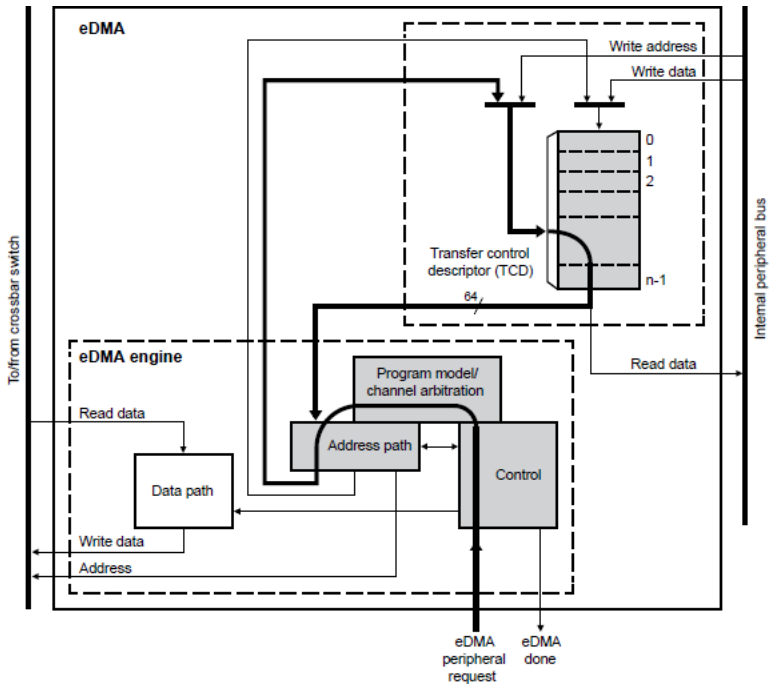

图4. 初始化传输

- **执行数据传输**
  - **数据移动**

如图5所示，地址路径、数据路径和控制单元等模块协同工作，以执行TCD中定义的实际数据传输。

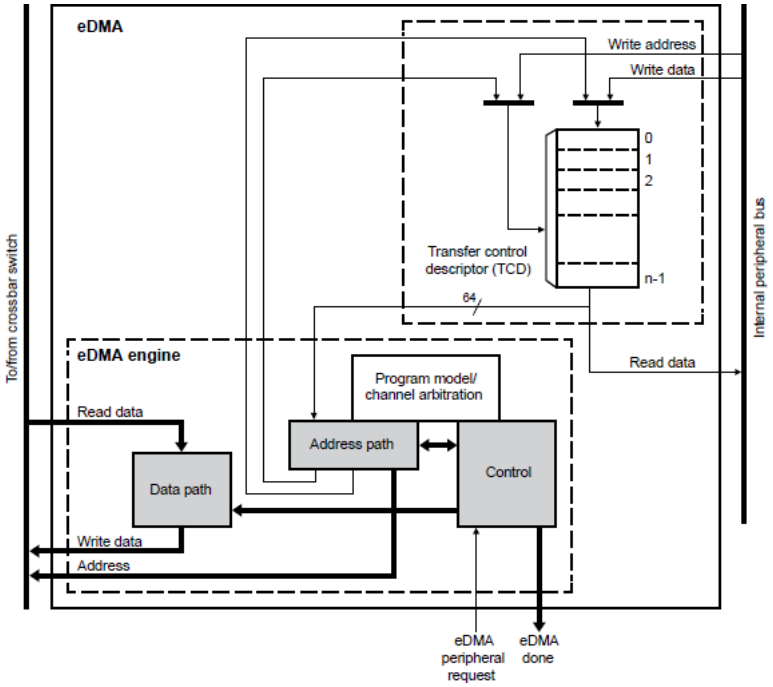

图5. eDMA数据传输

  - **源地址读取**

发起“源地址读取”，并在将获取的数据传输到目的地址之前将其临时存储在数据路径块中。

– **目的地址写入**

此“源地址读取/目的地址写入”循环一直持续到所有NBYTES的数据都传输完毕。

- **完成传输**
  - **TCD更新**

    如图6所示，在传输了NBYTES数据后，地址路径逻辑会根据操作更新TCD中的特定字段，如SADDR、DADDR和CITER。
  - **主迭代完成**

    如果主迭代计数达到其限制，则会进行更多的操作，包括最终地址调整和将BITER重新加载到CITER中。
  - **可选中断和分散/聚集**

    可能会发出一个可选的中断请求。如果启用了分散/聚集，则可以使用提供的指针从存储器中提取一个新的TCD。

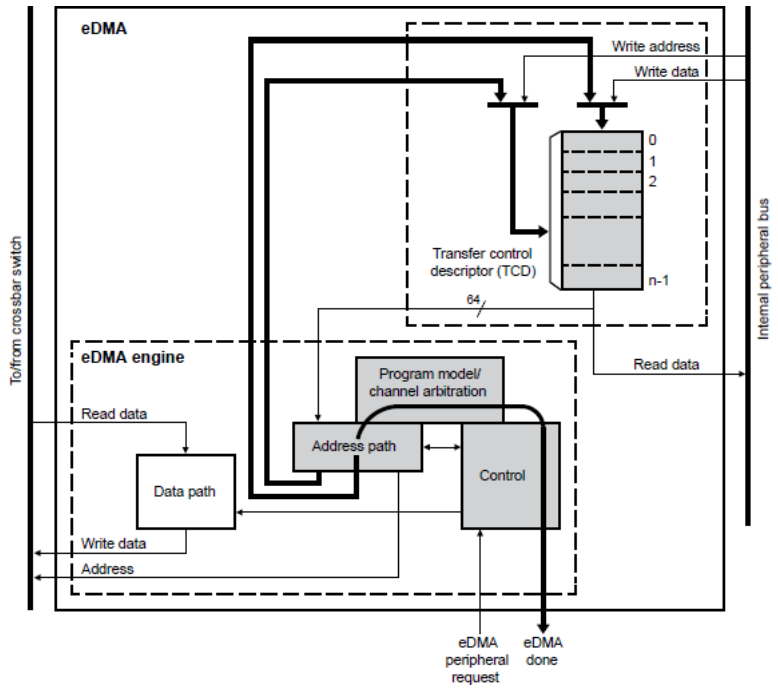

图6. 完成eDMA传输

2.1 通道激活

- 其它外设模块内发生的事件可以被启用以激活eDMA传输。在许多模块中，事件标志可以被置位，作为eDMA或中断请求。由于这些请求的源数量众多，实现了一个可配置的多路复用器（DMAMUX）来将外设DMA请求路由到DMA通道。
- 通道也可以通过软件激活。通道的TCD提供了一个START位，当该位被置位时会激活通道。这使得可以在软件中激活每个通道。START位还为测试和调试TCD提供了一个有用的工具，可以评估每次通道激活时是否执行了预期的传输。
- 通道链接提供了一种手段，使一个通道可以置位另一个通道的START位。链接的通道可以在传输的不同阶段或传输完成后激活。

2.2 通道仲裁

每个DMA通道的外设请求线由DMA多路复用器（DMAMUX）驱动。这是一种灵活的配置，允许用户选择适当的外设连接到DMA控制器的每个通道。DMA多路复用器用于将众多外设DMA源路由到各个DMA通道，它最多支持128个DMA请求信号（其中6个保留供将来使用）。有两个DMA多路复用器实体可用，每个实体都能将源路由到16个DMA通道。

表1. eDMA传输请求源

DMAMUX实体	DMA通道
0	0–15
1	0-15

DMA多路复用器（DMA_Mux）执行将外设DMA请求源路由到期望通道的任务。

表2. MCX Nx4x上的外设DMA请求

DMAMUX编号	DMAMUX0 - DMAMUX1（源描述符）	DMAMUX0 - DMAMUX1别名
0	Disabled（禁用）	—
1	Receive event（接收事件）	FlexSPI0
2	Transmit event（发送事件）	FlexSPI0
3-6	INT0 - INT3	PINT0
7-8至15-16	DMAREQ_M0 - DMAREQ_M1	CTIMER0至CTIMER4
17	Wake up event（唤醒事件）	WUU0
18	FIFO_request（FIFO请求）	MICFIL0
19-20	DMA0-DMA1	SCT0
21-22	FIFO A, FIFO B request（FIFO A、FIFO B请求）	ADC0
23-24	FIFO A, FIFO B request（FIFO A、FIFO B请求）	ADC1
25-27	FIFO_request（FIFO请求）	DAC0至DAC2
28-30	DMA_request（DMA请求）	CMP0至CMP2
31-32至37-38	OUT0A-OUT0B to OUT3A-OUT3B（OUT0A-OUT0B至OUT3A-OUT3B）	EVTG0
39-42	Req_Capt0 – Req_Capt3	PWM0
43-46	Req_val0 – Req_val3	PWM0
47-50	Req_capt0 – Req_capt3	PWM1
51-54	Req_val0 – Req_val3	PWM1
55, 56	—	保留
57, 58	Counter match event（计数器匹配事件）	LPTMR0至LPTMR1
59, 60	DMA request（DMA请求）	CAN0至CAN1
61-68	Shifter(0-7) Status DMA request OR Timer(0-7) Status DMA request（移位器(0-7)状态DMA请求或定时器(0-7)状态DMA请求）	FlexIO0
69-70至87-88	Receive and Transmit request（接收和发送请求）	LP_FLEXCOMM0至LP_FLEXCOMM9

表2. MCX Nx4x上的外设DMA请求（续）

DMAMUX编号	DMAMUX0 - DMAMUX1（源描述符）	DMAMUX0 - DMAMUX1别名
89、90	—	保留
91-92至93-94	Receive-Transmit request（接收-发送请求）	EVMSIM0至EVMSIM1
95-96至97-98	Receive-Transmit request（接收-发送请求）	I3C0至I3C1
99-100至101-102	Receive-Transmit request（接收-发送请求）	SAI0至SAI1
103-107	Ipd_req_sinc[0-4] or ipd_req_alt[0-4]（Ipd_req_sinc[0-4]或ipd_req_alt[0-4]）	SINC0
108-109至118-119	Pin event request 0-Pin event request 1（引脚事件请求0-引脚事件请求1）	GPIO0至GPIO5
120	End of Scan（扫描结束）	TSI0
121	Out of Range（超出范围）	TSI0

2.3 运行模式

eDMA支持两种模式，具体解释如下：

- **调试模式**：在此模式下，DMA通道被禁用。由于DMA通道主要通过DMA配置寄存器禁用和启用，此模式主要用作DMA通道在DMA通道多路复用器中的复位状态。它还可以用于在系统重新配置时临时挂起DMA通道。
- **正常模式**：在此模式下，DMA源被直接路由到指定的DMA通道。DMA多路复用器在此模式下的操作对系统是完全透明的。DMA源和目的（如外设结果或传输缓冲区）在准备好接收或传输数据时，请求/触发DMA传输。

3 传输过程

在配置eDMA之前，了解eDMA如何进行传输是非常有用的。

3.1 处理多个传输请求

在任何给定时间，只有一个通道可以主动执行传输。因此，为了处理多个待处理的传输请求，eDMA控制器提供了通道优先级排序功能。可以选择固定优先级或轮询优先级。

在固定优先级方案中，每个通道被分配一个优先级级别。当有多个请求待处理时，优先级最高的通道会首先执行其传输。默认情况下，实现的是固定优先级仲裁，为每个通道分配一个与其通道号相等的优先级。如有需要，可以分配其它优先级级别。高优先级通道可以抢占低优先级通道。当一个通道正在执行传输时，而一个传输请求被发送给更高优先级的通道，则会发生抢占。在这种情况下，低优先级通道会暂停其传输，并允许高优先级通道执行其传输。当高优先级通道完成其传输后，低优先级通道再恢复其传输。支持一个级别的抢占。抢占是一个选项，如有需要，必须按每个通道启用。

在轮询模式下，eDMA按顺序循环检查各个通道是否有待处理的请求。当到达有待处理请求的通道时，允许其执行传输。当传输完成后，eDMA继续循环检查各个通道，寻找下一个待处理的请求。

3.2 主传输循环和次传输循环

每次激活并执行一个通道时，都会从源地址向目的地址传输某些字节，即**NBYTES**。这被称为次传输循环。一个主传输循环由几个次传输循环组成。具体数量在TCD中指定。当次循环的迭代完成时，当前迭代（CITER）TCD字段会递减。当当前迭代字段用尽时，通道就完成了一个主传输循环。

图7所示为主循环和次循环之间的关系。在此示例中，通道被配置为一个主循环由一个次循环的三次迭代组成。次循环被配置为传输4个字节。

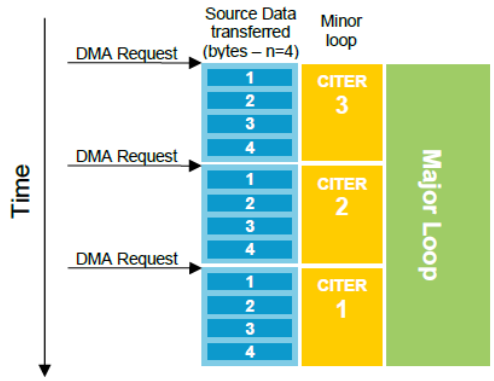

图7. 主传输循环和次传输循环

通道在完成每个次传输循环和主传输循环后会执行一系列任务。

3.3 完成次传输循环时

在完成次循环（不包括最后一个次循环）后，eDMA将执行以下任务：

- 递减当前迭代（CITER）计数器。
- 通过将当前源地址与有符号源地址偏移量相加来更新源地址：SADDR = SADDR + SOFF（随着传输的进行，源地址会自动更新。在完成次循环后，源地址包含了在次循环中读取的最后一组数据的源地址；偏移量将加到此值上）。
- 通过将当前目的地址与有符号目的地址偏移量相加来更新目的地址：DADDR = DADDR + DOFF。
- 更新通道状态位并请求（已启用）中断。
- 如果启用了通道链接，则在完成一个次要循环后将链接通道的启动位置位。

3.4 完成主传输循环时

在完成主循环（或最后一个次循环）后，eDMA执行以下操作：

- 通过将当前源地址与最后一个源地址调整值相加来更新源地址：SADDR = SADDR + SLAST
- 通过将当前目的地址与最后一个目的地址调整值相加来更新目的地址：DADDR = DADDR + DLAST
- 更新通道状态位并请求（已启用）中断
- 如果启用了通道链接，则在完成一个次要循环后将链接通道的启动位置位
- 从起始主迭代计数（BITER）字段重新加载当前迭代（CITER）

注：
有两种方法可以测试次循环的完成情况：

- ***硬件***
- ***软件***

TCD状态字段对硬件激活的通道执行以下序列：

表3. TCD状态字段

阶段	TCDn_CSR字段	CHn_CSR字段		状态
	START（启动）	ACTIVE（活动）	DONE（完成）	
1	0	0	0	通过硬件发起通道服务请求（外设请求已发出）。
2	0	1	0	通道正在执行。
3a	0	0	0	通道已完成次循环，处于空闲状态。
3b	0	0	1	通道已完成主循环，处于空闲状态。

在使用硬件发起的服务请求时，测试次循环是否完成的最佳方法是读取TCDn_CITER字段并测试其是否发生变化。

而在软件发起的服务请求中，TCDn_CSR[START]字段在阶段1时为**1**。

对于这两种激活类型，主循环完成状态都通过CHn_CSR[DONE]字段明确指示。无论通道如何激活，当通道开始执行时，TCDn_CSR[START]字段都会自动清零。

3.5 通道链接

通道链接（或链式传递）是一种机制，其中一个通道设置另一个通道（或自身）的TCDn_CSR[START]字段，从而发起对该通道的服务请求。当正确启用后，eDMA引擎会在主循环或次循环完成时自动执行此操作。

4 动态编程

此方法用于在通道执行期间更改编程模型。要更改组或通道的优先级级别，请执行以下步骤：

- 通过在CSR[HALT]字段写入**1**来暂停DMA。
- 根据需要更改组或通道的优先级。
- 通过在CSR[HALT]字段写入0来启用正常的DMA操作。

在执行动态通道链接请求时，请使用以下一致性模型。

1. 在TCDn_CSR[MAJORELINK]字段写入1。
2. 读回TCDn_CSR[MAJORELINK]字段。
3. 测试TCDn_CSR[MAJORELINK]请求状态：
   - 如果TCDn_CSR[MAJORELINK] = 1，则动态链接尝试成功。
   - 如果TCDn_CSR[MAJORELINK] = 0，则动态链接尝试未成功（通道已经处于退出状态）。

4.1 动态分散/聚集

分散/聚集是将新TCD自动加载到通道中的过程。它还允许DMA通道使用多个TCD，这可以：

- 使单个DMA通道能够使用多种数据传输配置。
- 允许将数据**分散**到多个目的地址或从多个源**聚集**数据。
- 在当前传输完成后，自动加载新的传输配置（TCD）。

由于配置可能会在执行过程中被更改，因此需要一个一致性模型。

为什么需要一致性?

- 如果用户希望通过启用TCDn_CSR[ESG]字段来执行动态分散/聚集操作，而同时eDMA引擎正在停止一个通道，则不清楚实际的分散/聚集请求是否得到了处理。

推荐解决方案

1. 完成后强制将TCDn_CSR[ESG]清零：
   - 每当用户写入TCDn_CSR时，应确保在通道完成其主循环（CHn_CSR[DONE] = 1）后，将TCDn_CSR[ESG]清零。
2. 在设置ESG之前清除DONE：
   - 要设置ESG，应首先清除DONE字段。

好处

- 无需多个DMA通道即可实现复杂的数据传输。
- 提高特定DMA操作的灵活性和效率。

5 配置eDMA

本节介绍了一些重要的配置步骤和寄存器字段。有关所有寄存器字段的详细信息，请参阅微控制器的参考手册。

5.1 配置步骤

要配置eDMA，请执行以下初始化步骤：

1. 对控制寄存器（MP_CSR）进行编程。此步骤仅在需要非默认配置时才必要。
2. 配置通道优先级寄存器（CHn_PRI）和组优先级级别寄存器（CHn_GRPRI）。此步骤仅在需要非默认配置时才必要。
3. 使用DMAEEI或 DMASEEI寄存器启用错误中断CHn_CSR[EEI]。此步骤仅在需要非默认配置时才必要。
4. 为所有使用的通道编写传输控制描述符（eDMA_TCDn）。如有必要，为分散/聚集机制配置TCD。
5. 配置适当的外设模块，并配置DMAMUX，以将激活信号路由到适当的通道。

5.2 传输控制描述符（TCD）

通道的所有传输属性都在该通道的唯一TCD中定义。每个TCD都存储在eDMA控制器的本地SRAM中。只有DONE、ACTIVE和STATUS字段在复位时被初始化。所有其它TCD字段在复位后都是未定义的，必须在通道被激活之前通过软件写入。如果不这样做，可能会导致通道的行为不可预测。图8所示为TCD的存储映射。

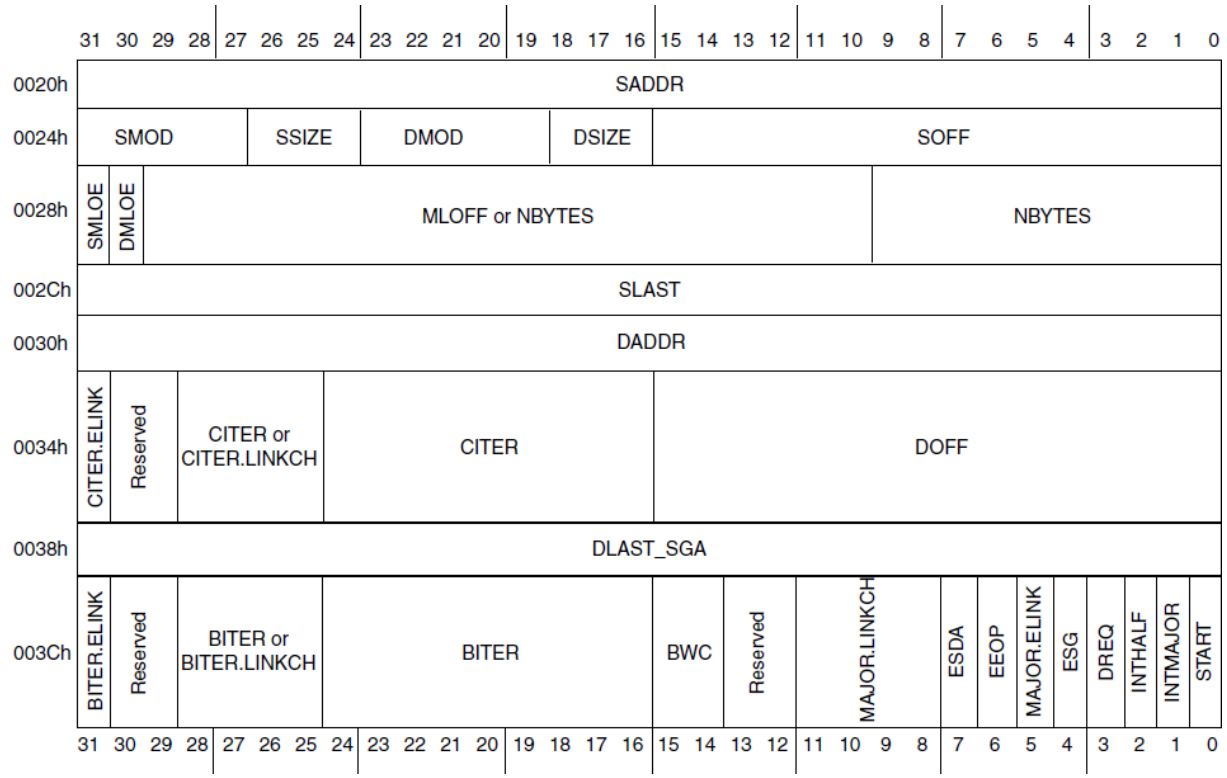

图8. TCD存储器映射

表4列出了TCD的元素及其功能。

表4. TCD字段说明

字段	说明
SADDR[31:0]	源地址 传输的源数据的存储地址。它允许选择存储映射的任何区域。当eDMA执行传输时，此字段会自动更新，以进行下一次传输。
SMOD[15:11]	源地址模数 它简化了环形数据队列的实现。当较低的地址字段恢复到其原始值，而较高的字段保持固定时，会创建一个环形缓冲区。 00000——禁用源地址模数功能。 xxxxx——允许递增的较低源地址位数。
SSIZE[10:8]	源数据传输大小 它定义了eDMA引擎的读取数据大小。它并未定义每次通道激活传输的数据量。 000——8位 001——16位 010——32位 011——保留 100——16字节突发 101——保留 110——保留

表4. TCD字段说明（续）

字段	说明
	111——保留
DMOD[7:3]	目的地址模数 它可用于实现环形数据目的地址。与SMOD相似，但针对的是目的地址。
DSIZE[2:0]	目的数据传输大小 它定义了eDMA引擎的数据写入大小。与SSIZE相似。
SOFF[15:0]	源地址有符号偏移量 在完成一个次循环后，加到当前源地址的有符号偏移量，以计算新的源地址值。
NBYTES[29:0]/[9:0]	通道的每个服务请求中要传输的字节数。 每次激活通道时要传输的字节数。 该字段的长度取决于是否启用/禁用次偏移量。
SMLOE[31]	源地址次循环偏移量启用 0——次循环偏移量不应用于SADDR。 1——次循环偏移量应用于SADDR。
DMLOE[30]	目的地址次循环偏移量启用 0——次循环偏移量不应用于DADDR。 1——次循环偏移量应用于DADDR。
MLOFF[29:10]	如果设置了SMLOE或DMLOE，则此字段表示应用于源地址或目的地址的有符号扩展偏移量，以在次循环完成后形成下一个状态值。
SLAST_SDA[31:0]	上一次源地址调整量 在主循环完成后添加到源地址的有符号偏移量，以计算新的源地址值。它可用于将源地址恢复为原始值，或将源地址调整为下一个数据结构。
DADDR[31:0]	目的地址 指向目的数据的存储地址。
CITER_ELINK	在次循环完成后启用通道链接 0——禁用“在次循环完成后进行通道链接”。 1——启用“在次循环完成后进行通道链接”。 ***注：*** *此字段必须等于BITER_E_LINK字段，否则将报告配置错误。*
LINKCH[5:0]	次循环链路通道号 如果启用了通道到通道链接（ELINK = 1），那么在次循环用尽后，eDMA引擎通过在该通道的TCDn_CSR[START]字段写入**1**，向该字段定义的通道发起通道服务请求。
CITER[14:0] or CITER[8:0]	当前迭代计数 此9位（ELINK = 1）或15位（ELINK = 0）计数表示通道的当前主循环计数。每次通道完成服务请求时，它都会递减，并被写回TCD存储器。在主要迭代计数用尽后，通道执行一系列操作，例如最终源地址和目的地址计算，并在从起始迭代计数（BITER）字段重新加载CITER字段之前，可选择生成一个中断以指示通道完成。
DOFF[15:0]	目的地址有符号偏移量 在完成一个次循环后，加到当前目的地址的有符号偏移量，以计算下一个目的地址。
DLAST_SGA[31:0]	上一次目的地址调整量或下一个TCD的存储地址

表4. TCD字段说明（续）

字段	说明
	如果禁用分散/聚集（ESG = 0），则此字段中的值执行与目的地址的SLAST字段相同的任务。 如果启用分散/聚集（ESG = 1），则此字段用作指向包含此通道下一个TCD的0-modulo-32区域的指针。
ELINK	启用链接 0b——禁用通道到通道链接。 1b——启用通道到通道链接。
LINKCH[12:9]	链接通道号 如果启用了通道到通道链接（ELINK = 1），则在次循环用尽后，eDMA引擎通过设置该通道的TCDn_CSR[START]字段，在该字段定义的通道上发起通道服务请求。
BITER[14:0] or BITER[8:0]	起始主迭代计数 由于传输控制描述符首先由软件加载，必须将此9位（ELINK = 1）或15位（ELINK = 0）字段设置为与CITER字段中的值相等。当主要迭代计数用尽时，eDMA会将此字段的内容重新加载到CITER字段中。如果通道配置为执行单个服务请求，则BITER和CITER的初始值必须为0x0001。
BWC[15:14]	带宽控制 它提供了一种控制eDMA使用的总线带宽量的方法。 00——无暂停-占用全部带宽。 01——保留。 10——eDMA在每次读/写后暂停4个周期。 11——eDMA在每次读/写后暂停8个周期。
MAJORLINKCH[11:8]	主循环链接通道号 如果（MAJORELINK = 0），则： 在主循环计数器用尽后，不会执行通道到通道链接。 否则： 在主循环计数器用尽后，eDMA引擎发起通道服务请求。
ESDA	启用“存储目的地址” 0b——禁用“将目的地址存储到系统内存”功能。 1b——启用“将目的地址存储到系统内存”功能。
EEOP	启用“数据包结束处理” 0b——禁用“数据包结束操作”。 1b——启用“数据包结束硬件输入信号”。
MAJORLINK[5]	在主循环完成后启用通道链接 0——禁用“在主循环完成后进行通道链接”。 1——启用“在主循环完成后进行通道链接”。
ESG[4]	启用分散/聚集处理 0——禁用分散/聚集处理。 1——启用分散/聚集处理。
DREQ[3]	禁用请求 如果在通道完成主要循环时设置了此位，则eDMA会清除相应的DMAERQ，从而禁用传输请求。 0——通道的DMAERQ位不受影响。 1——通道的DMAERQ位在主要循环完成后被清除。

表4. TCD字段说明（续）

字段	说明
INTHALF[2]	当主循环完成一半时生成中断。当CITER = BITER ÷ 2时，eDMA在DMAINT寄存器中发出中断请求。 0——禁用“主循环完成一半时生成中断”。 1——启用“主循环完成一半时生成中断”。
INTMAJOR[1]	在主循环完成后生成中断。当CITER = 0时，eDMA在DMAINT寄存器中发出中断请求。 0——禁用“主循环完成后生成中断”。 1——启用“主循环完成后生成中断”。
START[0]	通道启动 在此位写入**1**会显式激活通道，并执行次要循环传输。

如果通道的TCD描述符被配置了非法值或某些值的非法组合，则会在DMAERR寄存器中报告通道错误。

6 eDMA配置示例

本节详细介绍了基于FRDM-MCXN947 SDK的两个eDMA配置示例，从一个简单的配置开始，然后在应用层级逐步深入到eDMA的更高级特性和功能：

1. 基本传输（包括通道链接）
2. 分散/聚集

6.1 配置示例1：基本传输

此示例配置eDMA执行基本的软件触发eDMA传输，并在第一个通道的主循环完成后进行通道链接。

6.1.1 要求

在内部SRAM中创建了四个数据数组。第一个数组`srcAddr[]`的大小为4字节，而其它三个数组`destAddr0[]`、`destAddr1[]`和`destAddr3[]`也都是4字节，用于演示TCD的不同设置。

- AT_NONCACHEABLE_SECTION_INIT(uint32_t srcAddr[BUFFER_LENGTH]) = {0x01U, 0x02U, 0x03U, 0x04U}
- AT_NONCACHEABLE_SECTION_INIT(uint32_t destAddr0[BUFFER_LENGTH]) = {0x00U, 0x00U, 0x00U, 0x00U}
- AT_NONCACHEABLE_SECTION_INIT(uint32_t destAddr1[BUFFER_LENGTH]) = {0x00U, 0x00U, 0x00U, 0x00U}
- AT_NONCACHEABLE_SECTION_INIT(uint32_t destAddr2[BUFFER_LENGTH]) = {0x00U, 0x00U, 0x00U, 0x00U}

当执行传输的通道被软件激活时，序列中的第一个32位数据从源地址移动到目的地址。

在主循环完成时，通过通道链接自动触发下一个通道，并由第二个通道移动32位，依此类推。

当此传输完成后，该通道将不再被使用，因此无需为将来的传输恢复或准备该通道。

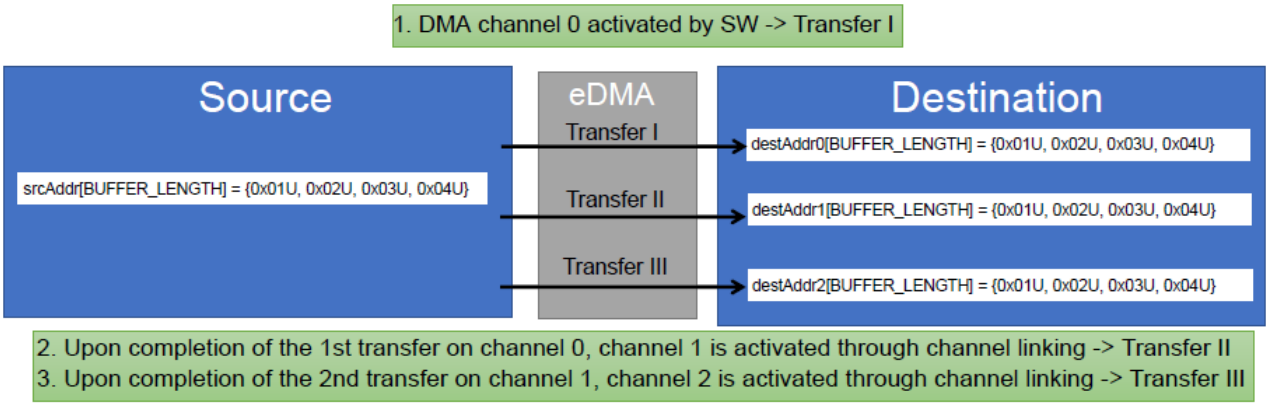

图9. 示例1的要求

6.1.2 模块配置

此示例使用软件通道激活第一个TCD，并通过通道链接激活第二个TCD和第三个TCD。不需要配置DMAMUX或eDMA模块寄存器。只需在通过TCD配置和激活通道之前加载源数据。

下面给出了在通道0、1和2上配置TCD的代码（包括在通道1和2上启用通道链接）：

```
/* Configure CH0 */
DMA_0.TCD[0].SADDR = (int)&srcAddr[0];
/*DMA_0.TCD[0].ATTR = 0x02 which is defining SMOD, SSIZE, DMOD & DSIZE as
 below:*/

DMA_0.TCD[0].SMOD = 0;
DMA_0.TCD[0].SSIZE = 0x2;         /* 32-bit */
DMA_0.TCD[0].DMOD = 0;
DMA_0.TCD[0].DSIZE = 0x2;         /* 32-bit */
DMA_0.TCD[0].SOFF = 0x4;
/*TCD0_NBYTES_MLOFFNO & TCD0_NBYTES_MLOFFYES defines TCD Transfer Size without
 or with minor loop offset where SMLOE, DMLOE are disabled*/
DMA_0.TCD[0].NBYTES = 8;
DMA_0.TCD[0].SLAST_SDA  = 0;
DMA_0.TCD[0].DADDR = 0x20000030;
/*TCD0_CITER_ELINKNO & TCD0_CITER_ELINKYES which indicates the Current Major
 loop channel linking enabled or disabled:*/

DMA_0.TCD[0].ELINK = 1;           /*Enable Link*/
DMA_0.TCD[0].LINKCH = 1;
DMA_0.TCD[0].CITER = 2;           /*Current Major iteration count*/
DMA_0.TCD[0].DOFF = 0x4;
DMA_0.TCD[0].DLAST_SGA = 0;
/*TCD0_BITER_ELINKNO & TCD0_BITER_ELINKYES which indicates the Beginning Major
 loop channel linking enabled or disabled:*/

DMA_0.TCD[0].ELINK = 1;            /*Enable Link*/
DMA_0.TCD[0].LINKCH = 1;
DMA_0.TCD[0].BITER = 2;            /*Beginning Major iteration count*/
/*DMA_0.TCD[0].CSR as shown below: */
DMA_0.TCD[0].BWC = 0;
DMA_0.TCD[0].MAJORLINKCH = 1;       /* Link to channel 1 */
```

```
DMA_0.TCD[0].MAJORELINK = 1;         /* Enable channel linking*/
DMA_0.TCD[0].ESG = 0;
DMA_0.TCD[0].DREQ = 1;
DMA_0.TCD[0].INTHALF = 0;
DMA_0.TCD[0].INTMAJ = 0;

/* Configure CH1 */
DMA_0.TCD[1].SADDR = (int)&srcAddr[0];
/*DMA_0.TCD[1].ATTR = 0x02 which is defining SMOD, SSIZE, DMOD & DSIZE as
 below:*/
DMA_0.TCD[1].SMOD = 0;
DMA_0.TCD[1].SSIZE = 0x2;             /* 32-bit */
DMA_0.TCD[1].DMOD = 0;
DMA_0.TCD[1].DSIZE = 0x2;             /* 32-bit */
DMA_0.TCD[1].SOFF = 0x4;
/*TCD1_NBYTES_MLOFFNO & TCD1_NBYTES_MLOFFYES defines TCD Transfer Size without
 or with minor loop offset where SMLOE, DMLOE are disabled*/
DMA_0.TCD[1].NBYTES = 16;             /* 4x32-bits */
DMA_0.TCD[1].SLAST_SDA  = 0;
DMA_0.TCD[1].DADDR = 0x20000040;
/*TCD1_CITER_ELINKNO & TCD1_CITER_ELINKYES which indicates the Current Major
 loop channel linking enabled or disabled:*/

DMA_0.TCD[1].ELINK = 0;               /*Enable Link*/
DMA_0.TCD[1].LINKCH = 0;
DMA_0.TCD[1].CITER = 1;               /*Current Major iteration count*/
DMA_0.TCD[1].DOFF = 0x4;
DMA_0.TCD[1].DLAST_SGA = 0;
/*TCD1_BITER_ELINKNO & TCD1_BITER_ELINKYES which indicates the Beginning Major
 loop channel linking enabled or disabled:*/

DMA_0.TCD[1].ELINK = 0;                /*Enable Link*/
DMA_0.TCD[1].LINKCH = 0;
DMA_0.TCD[1].BITER = 1;                /*Beginning Major iteration count*/
/*DMA_0.TCD[1].CSR as shown below:*/
DMA_0.TCD[1].BWC = 0;
//DMA_0.TCD[1].MAJORLINKCH = 0;
DMA_0.TCD[1].MAJORELINK = 0;
DMA_0.TCD[1].ESG = 0;
DMA_0.TCD[1].DREQ = 1;
DMA_0.TCD[1].INTHALF = 0;
DMA_0.TCD[1].INTMAJ = 0;
/* Configure CH2 */
DMA_0.TCD[2].SADDR = (int)&data_array1[0];
/*DMA_0.TCD[2].ATTR = 0x02 which is defining SMOD, SSIZE, DMOD & DSIZE as
 below:*/
DMA_0.TCD[2].SMOD = 0;
DMA_0.TCD[2].SSIZE = 0x2;               /* 32-bit */
DMA_0.TCD[2].DMOD = 0;
DMA_0.TCD[2].DSIZE = 0x2;               /* 32-bit */
DMA_0.TCD[2].SOFF = 0x4;
/*TCD2_NBYTES_MLOFFNO & TCD2_NBYTES_MLOFFYES defines TCD Transfer Size without
 or with minor loop offset where SMLOE, DMLOE are disabled*/
DMA_0.TCD[2].NBYTES = 16;               /* 4x32-bits */
DMA_0.TCD[2].SLAST_SDA  = 0;
DMA_0.TCD[2].DADDR = 0x20000050;
/*TCD2_CITER_ELINKNO & TCD2_CITER_ELINKYES which indicates the Current Major
 loop channel linking enabled or disabled: */

DMA_0.TCD[2].ELINK = 0;                  /*Enable Link*/
```

```
DMA_0.TCD[2].LINKCH = 0;
DMA_0.TCD[2].CITER = 1;                    /*Current Major iteration count*/
DMA_0.TCD[2].DOFF = 0x4;
DMA_0.TCD[2].DLAST_SGA = 0;
/*TCD2_BITER_ELINKNO & TCD2_BITER_ELINKYES which indicates the Beginning Major
 loop channel linking enabled or disabled:*/

DMA_0.TCD[2].ELINK = 0;                     /*Enable Link*/
DMA_0.TCD[2].LINKCH = 0;
DMA_0.TCD[2].BITER = 1;                     /*Beginning Major iteration count*/
/*DMA_0.TCD[2].CSR as shown below:*/
DMA_0.TCD[2].BWC = 0;
//DMA_0.TCD[2].MAJORLINKCH = 0;
DMA_0.TCD[2].MAJORELINK = 0;
DMA_0.TCD[2].ESG = 0;
DMA_0.TCD[2].DREQ = 1;
DMA_0.TCD[2].INTHALF = 0;
DMA_0.TCD[2].INTMAJ = 0;
```

注：*被注释掉的位域也显示出来，以便可以查看所有TCD字段。如果一个位域被注释掉，则其值被设置为0。*

通过设置MAJORELINK = 1来启用通道链接。

第一次DMA传输通过设置TCD的启动位来发起：

```
DMA_0.TCD[0].START = 1; /* Start transfer on channel 0 */
```

通道1通过通道链接自动启动。

如果可能，在调试环境中单步执行代码，并在通道激活和传输执行时监控源和目的存储地址。在主要循环完成后，源地址和目的地址将恢复。因此，进一步激活通道0会导致传输过程重复。

在此配置下，每次传输一个32位值时，完成一个次循环。当所有传输都完成后，主循环也随之完成。

6.2 配置示例2：分散/聚集

此示例配置eDMA执行软件触发的eDMA传输，并在第一个主循环完成后使用分散/聚集机制。

6.2.1 要求

在内部SRAM中创建了两个数据数组和一个TCD数组。第一个数组`srcAddr`的大小为8字节，而`destAddr[]`也是8字节，用于演示TCD的不同设置。

`tcdMemoryPoolPtr`结构体包含新的TCD内容，这些内容需要在完成第一个主循环后加载到通道的TCD中。

当执行传输的通道由软件激活时，序列中的第一个32位数据从源地址移动到目的地址。

在第一个主循环完成后，执行分散/聚集，并将`tcdMemoryPoolPtr`结构体中包含的数据加载到通道的TCD中。当新的TCD加载时，由于设置了START位，下一次传输会自动触发。当第二次传输完成后，该通道不再被使用，因此无需为将来的传输恢复或准备该通道。

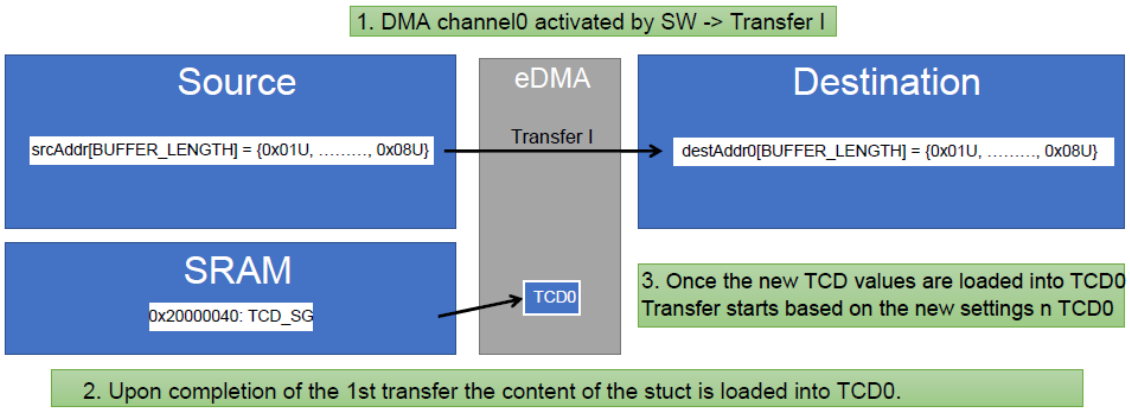

图10. 示例2的要求

6.2.2 模块配置

此示例使用软件通道激活进行第一次传输。当新的TCD加载时，由于设置了START位时，第二次传输会自动开始。无需配置DMA_Mux或eDMA模块寄存器。只需在通过TCD配置和激活通道之前加载源数据。

在通道0上执行传输的代码（包括启用分散/聚集）如下所示：

```
/* Configure CH0 */
DMA_0.TCD[0].SADDR = (int)&srcAddr[0];
DMA_0.TCD[0].SMOD = 0;
DMA_0.TCD[0].SSIZE = 0x2;         /* 32-bit */
DMA_0.TCD[0].DMOD = 0;
DMA_0.TCD[0].DSIZE = 0x2;         /* 32-bit */
DMA_0.TCD[0].SOFF = 0x4;
DMA_0.TCD[0].NBYTES = 16;         /* 2 structures of 16 bytes each will be
 created */
DMA_0.TCD[0].SLAST_SDA = 0;
/*TCD0_CITER_ELINKNO & TCD0_CITER_ELINKYES which indicates the Current
Major loop channel linking enabled or disabled: */

DMA_0.TCD[0].ELINK = 0;            /*Enable Link*/
DMA_0.TCD[0].LINKCH = 0;
DMA_0.TCD[0].CITER = 1;            /*Current Major iteration count*
DMA_0.TCD[0].DADDR = 0x20000040;
DMA_0.TCD[0].DOFF = 0x4;
DMA_0.TCD[0].DLAST_SGA = 0x20000020;
/*TCD0_BITER_ELINKNO & TCD0_BITER_ELINKYES which indicates the Beginning
Major loop channel linking enabled or disabled: */

DMA_0.TCD[0].ELINK = 0;             /*Enable Link*/
DMA_0.TCD[0].LINKCH = 0;
DMA_0.TCD[0].BITER = 1;             /*Beginning Major iteration count*/
/*DMA_0.TCD[0].CSR as shown below: */

/*For the transfer of 1st structure the register values will be*/
DMA_0.TCD[0].BWC = 0;
DMA_0.TCD[0].MAJORLINKCH = 0;
DMA_0.TCD[0].MAJORELINK = 0;        /* Disable channel linking */
DMA_0.TCD[0].ESG = 0;               /* Enable Sgatter/gather */
DMA_0.TCD[0].DREQ = 1;
```

```
DMA_0.TCD[0].INTHALF = 0;
DMA_0.TCD[0].INTMAJ = 0;

/*And for the second structure the values will be: */
DMA_0.TCD[0].BWC = 0;
DMA_0.TCD[0].MAJORLINKCH = 0;
DMA_0.TCD[0].MAJORELINK = 0;        /* Disable channel linking */
DMA_0.TCD[0].ESG = 1;               /* Enable Sgatter/gather */
DMA_0.TCD[0].DREQ = 0;
DMA_0.TCD[0].INTHALF = 0;
DMA_0.TCD[0].INTMAJ = 2;
```

注：*被注释掉的位域也显示出来，以便查看所有TCD字段，并使差异更加明显。*

注：*用于分散/聚集的TCD数组的32字节必须按32字节对齐。通道重新加载在主迭代计数完成时执行。分散/聚集地址必须为0-modulo-32-byte；否则会报告配置错误。*

`tcdMemoryPoolPtr`结构体是根据TCD存储器映射进行配置的（比较图7）。数组的第一个元素包含源地址（SADDR），然后指向要传输的数据数组。最后一个字被配置为设置START位，以便在加载新的TCD时开始传输。

如果可以，请在调试环境中单步执行代码，并在通道激活和传输执行时监控源和目的存储地址。在第一个主循环完成后，`tcdMemoryPoolPtr`数组中定义的值被加载到`TCD[0]`中。

7 调试技巧

虽然本应用笔记为使用eDMA提供了一个坚实的基础，但eDMA是一个功能强大的模块，有许多可能的使用场景，无法在本应用笔记中全部涵盖。为了帮助开发人员在开发应用程序时调试可能出现的问题，eDMA包含了错误状态（ES）寄存器，可以作为诊断DMA传输问题的强大工具。

通过有效地使用这些寄存器并理解它们的位域，开发人员可以：

- **主动识别和解决错误**：基于特定位标志的及时错误处理可以防止数据损坏和系统崩溃。
- **准确定位问题根源**：区分主设备错误和特定通道的错误，可以实现有针对性的调试和更快的问题解决。
- **验证DMA配置**：检查诸如无效的传输大小或描述符表问题等错误，有助于确保正确的设置。
- **增强应用程序的稳健性**：结合基于这些寄存器的错误处理例程，可以提高应用程序的可靠性和稳定性。

MP_ES（主设备错误状态）寄存器：提供有关影响整个eDMA控制器而非特定通道的错误的深入观察。

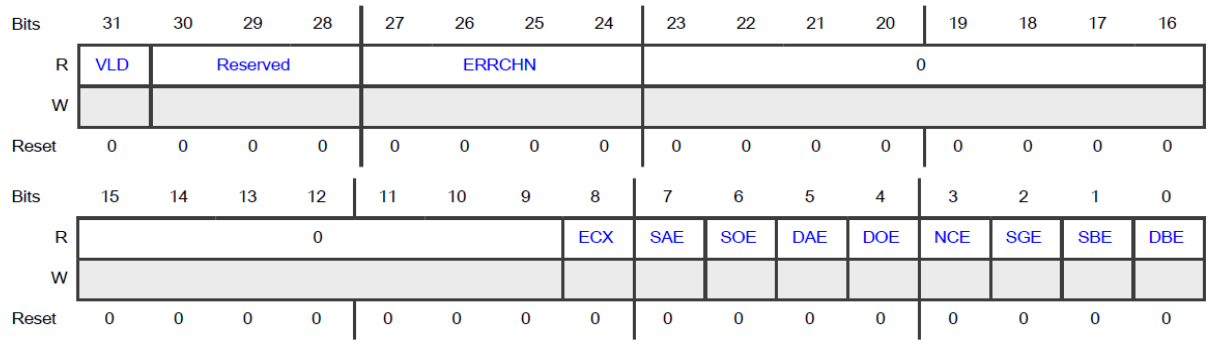

图11. MP_ES（主设备错误状态）寄存器

关键位域：

表5. 错误状态寄存器字段说明

字段	说明
VLD	CH_ES[ERR]状态位，指示其中一个通道出现了错误。通过查看“通道错误状态”寄存器可以确定确切的通道号。
ERRCHN	错误通道号
ECX	通过错误取消传输位DMA_CR[ECX]取消了传输。
SAE	TCD_SADDR字段中的配置错误（与TCD_ATTR[SSIZE]不一致）
SOE	TCD_SOFF字段中的配置错误（与TCD_ATTR[SSIZE]不一致）
DAE	TCD_DADDR字段中的配置错误（与TCD_ATTR[DSIZE]不一致）
DOE	TCD_DOFF字段中的配置错误（与TCD_ATTR[DSIZE]不一致）
NCE	TCD_NBYTES或TCD_CITER字段中的配置错误。初始时，TCD_CITER必须被编程为与TCD_BITER相同的值。
SGE	分散/聚集错误，这表明TCD_DLAST_SGA字段中存在配置错误，当分散/聚集功能启用时（TCD_CSR[ESG] = 1），该字段必须位于32字节边界上。
SBE	源地址读取时发生总线错误。
DBE	目的地址写入时发生总线错误。

通道错误状态寄存器：捕获特定于单个通道的错误，有助于准确定位有问题的传输。

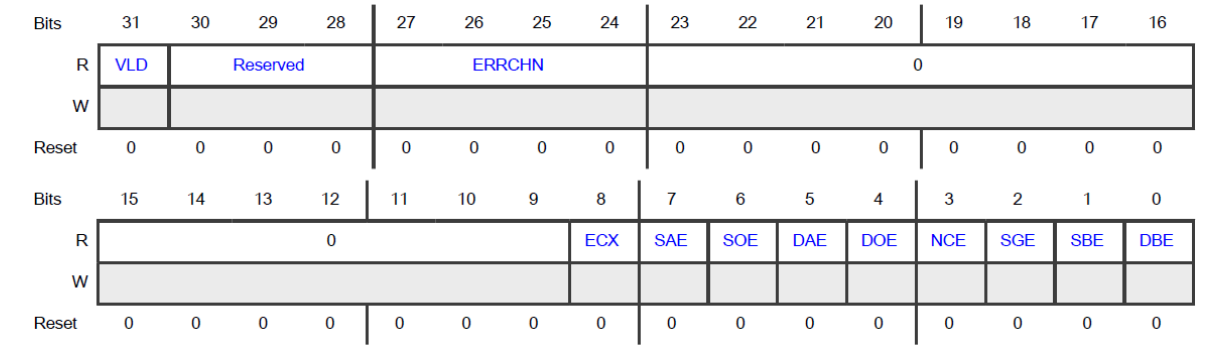

图12. 通道错误状态寄存器

表6. 错误状态寄存器字段说明

字段	说明
ERR	如果通道中发生错误，则启用。
SAE	TCD_SADDR字段中的配置错误（与TCD_ATTR[SSIZE]不一致）。
SOE	TCD_SOFF字段中的配置错误（与TCD_ATTR[SSIZE]不一致）
DAE	TCD_DADDR字段中的配置错误（与TCD_ATTR[DSIZE]不一致）
DOE	TCD_DOFF字段中的配置错误（与TCD_ATTR[DSIZE]不一致）
NCE	TCD_NBYTES或TCD_CITER字段中的配置错误。初始时，TCD_CITER必须被编程为与TCD_BITER相同的值。
SGE	分散/聚集错误，这表明TCD_DLAST_SGA字段中存在配置错误，当分散/聚集功能启用时（TCD_CSR[ESG] = 1），该字段必须位于32字节边界上。
SBE	源地址读取时发生总线错误。
DBE	目的地址写入时发生总线错误。

当DMA事务中发生错误时，它会根据错误类型在此ES寄存器中设置相应的标志位。下表列出了ES所指示的错误。

表7. 错误状态寄存器字段说明

字段	说明
VLD	CH_ES[ERR]状态位，指示其中一个通道出现了错误。通过查看“通道错误状态”寄存器可以确定确切的通道号。
ERRCHN	错误通道号。
ECX	通过错误取消传输位DMA_CR[ECX]取消了传输。
SAE	TCD_SADDR字段中的配置错误（与TCD_ATTR[SSIZE]不一致）。
SOE	TCD_SOFF字段中的配置错误（与TCD_ATTR[SSIZE]不一致）
DAE	TCD_DADDR字段中的配置错误（与TCD_ATTR[DSIZE]不一致）
DOE	TCD_DOFF字段中的配置错误（与TCD_ATTR[DSIZE]不一致）
NCE	TCD_NBYTES或TCD_CITER字段中的配置错误。初始时，TCD_CITER必须被编程为与TCD_BITER相同的值。
SGE	分散/聚集错误，这表明TCD_DLAST_SGA字段中存在配置错误，当分散/聚集功能启用时（TCD_CSR[ESG] = 1），该字段必须位于32字节边界上。
SBE	源地址读取时发生总线错误。
DBE	目的地址写入时发生总线错误。

调试时另一个有用的工具是软件启动位TCD_CSR[START]。它可以用于在软件中启动任何配置，并且是检查配置是否按预期运行的好方法。

8 结论

MCx Nx4x系列微控制器中的eDMA技术为各种应用解锁了显著的性能优势。本应用笔记提供了对MCX Nx4x eDMA控制器的深入理解和实用知识。它使用户能够通过有效地使用这些功能，来创建适合不同应用的eDMA配置。开发人员可以最大限度地提高数据吞吐量，最小化CPU负载，并创建高性能的嵌入式系统。随本应用笔记提供的源代码可以用作配置的基础。

有关恩智浦MCX Nx4x系列的更多信息，请访问nxp.com.cn。

9 关于本文中源代码的说明

本文中所示的示例代码具有以下版权和BSD-3-Clause许可：

2024年恩智浦版权所有；在满足以下条件的情况下，可以源代码和二进制文件的形式重新分发和使用本源代码（无论是否经过修改）：

1. 重新分发源代码必须保留上述版权声明、这些条件和以下免责声明。
2. 以二进制文件形式重新分发时，必须在文档和/或随分发提供的其他材料中复制上述版权声明、这些条件和以下免责声明。
3. 未经事先书面许可，不得使用版权所有者的姓名或参与者的姓名为本软件的衍生产品进行背书或推广。

本软件由版权所有者和参与者“按原样”提供，不承担任何明示或暗示的担保责任，包括但不限于对适销性和特定用途适用性的暗示保证。在任何情况下，无论因何种原因或根据何种法律条例，版权所有者或参与者均不对因使用本软件而导致的任何直接、间接、偶然、特殊、惩戒性或后果性损害（包括但不限于采购替代商品或服务；使用损失、数据损失或利润损失或业务中断）承担责任，无论是因合同、严格责任还是侵权行为（包括疏忽或其他原因）造成的，即使事先被告知有此类损害的可能性也不例外。

10 修订历史

表8汇总了本文的修订情况。

表8. 修订历史

文档ID	发布日期	说明
AN14300 v.1.0	2024年9月23日	首次公开发布

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.
Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com.cn/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

目录