

运用 **MSPM0** 的 **DMA** 和 **Event** 功能实现 **ADC** 多路周期性采样

Leon Yan

China Central FAE Team

ABSTRACT

全新一代的 MSPM0 微控制器基于 Cortex M0+内核，主频覆盖 24MHz 到 80MHz，提供了丰富的配置和多种增强功能，在内部的模拟接口上，最高达到 4MHz 采样率的 12bit ADC，以及 12bit 的 1MHz DAC，为数据采集和模拟输出提供了高性能的保障。多个内部集成的 OPA 运放节省了外部器件，降低了 PCB 空间占有率。内部的多路 DMA 通道，为多个外设、存储的数据传输提供了高速通路，降低 MCU 内核的占有率。新引入的 Event 事件系统带来了更加灵活的事件 Post 和 Subscribe 机制，对于各个模块、外设的相互关联带来了更多的可选择性和灵活性。

本应用文档中针对多路信号采样这类应用，运用 MSPM0G3507 的 DMA 和 Event 功能，在尽可能减少 MCU 内核被中断的情况下，实现 ADC 多路周期性采样及数据传输。

Contents

1. ADC 多路周期性采样需求简介	3
1.1 MSPM0 ADC 概述	3
1.2 MSPM0 DMA 概述	4
1.3 MSPM0 Event 概述	6
1.3.1 事件路由结构 (Event Fabric Routing)	6
1.3.2 事件路由表 (Event Routing Map)	7
2. 使用 Timer 进行周期性 ADC 采样触发	8
2.1 MSPM0 Timer 概述	8
3. 利用 Event 驱动 DMA 进行 ADC 数据搬移	9
4. 使用 FIFO 以优化 DMA 的搬移效率	11
5. 使用双缓冲以留足时间余量给 CPU 进行数学运算处理	13
6. 总结	14

Figures

图 1-1. 多路 ADC 采样框图	3
图 1-2. ADC 结构框图	4
图 1-3. DMA 结构框图	5
图 1-4. 静态事件路由 (Static Event Route)	6
图 1-5. DMA 事件路由 (DMA Event Route)	7
图 1-6. 通用事件路由 (Generic Event Route)	7
图 1-7. 通用事件路由 (Generic Event Route)	7
图 1-8. 事件路由表 (Event Routing Map)	8
图 2-1. 周期性定时器 (Repeat Down Counting Mode)	9
图 2-2. Timer G 事件配置	9
图 2-3. ADC 事件配置	9
图 3-1. ADC Auto Sampling 模式	10
图 3-2. ADC 采样窗口配置	10
图 3-3. ADC 采样窗口 Timer 配置	10
图 3-4. ADC DMA 内部接口	11
图 3-5. ADC DMA 触发源配置	11
图 4-1. ADC 采样模式和触发条件配置表	12
图 4-2. ADC 采样触发条件配置	12
图 4-3. ADC FIFO 模式下 DMA 触发条件配置	13

1 ADC 多路周期性采样需求简介

在一些应用场合下，我们会遇到针对多路模拟信号的采样需求，比如电压、电流的采样，一般需要周期性的用 ADC 进行数据采集，然后由 CPU 对得到的数据进行一系列运算。通常情况下，这种周期性的采样意味着 CPU 要周期性的控制 ADC 采样转换过程，周期性的去读取采样结果，对整个系统来讲 CPU 占用率过高，采样的周期性容易被打断，实际应用过程中往往利用 DMA 来减轻 CPU 数据搬移的负担。本应用文档利用 MSPM0 的 DMA 和 Event 功能，更加灵活的实现了此功能，大大减少了 CPU 的中断频率，可以更好的控制功耗，降低 CPU 负载。

这里我们针对 8 通道模拟信号进行采样，其采样周期为 1ms，每次通过顺序采样模式完成对 8 通道信号的采样，之后通过 DMA 将 ADC 的转换结果搬到 RAM 中的 Buffer1，在经过 64 次采样后填满 Buffer0，然后中断 CPU 切换 DMA 的目标地址到 Buffer1，依次轮询。

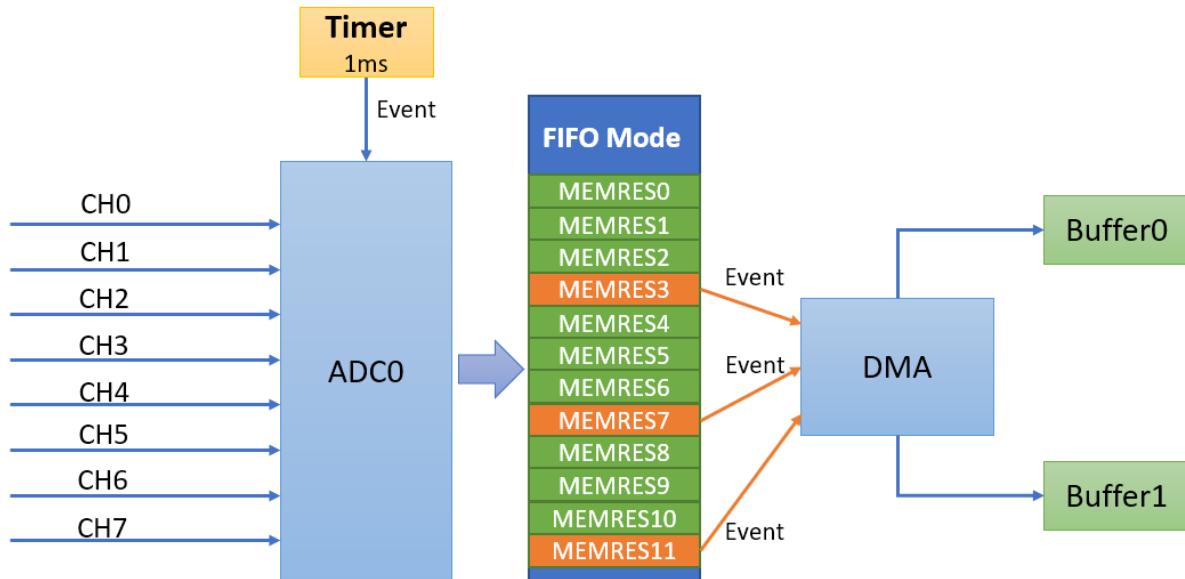


图 1-1. 多路 ADC 采样框图

1.1 MSPM0 ADC 概述

MSPM0 内部集成的 SAR ADC 可以支持 8bit、10bit、12bit 的采样精度，最高采样率可以达到 4Msps，最多支持到 12 个结果寄存器，以尽可能在 ADC 进行采样、转换的过程中避免中断 CPU。

MSPM0 ADC 特点：

- 12 bits 精度下的最高采样率可以达到 4-Msps
- 集成了硬件平均计算单元，最大做到 128 个结果的平均计算
- 两种 ADC 采样触发源：软件触发和 Event 触发
- 最多支持到 16 个 ADC 输入通道
- 支持 Single-channel, repeat-single-channel, sequence, repeat-sequence 四种转换模式
- 12 个 ADC 转换结果寄存器(MEMRES0:11)，同时也可以作为 FIFO 使用
- ADC 转换结果窗口阈值比较功能，降低 CPU 负载，降低功耗
- 支持 ADC 对 DMA 的多个触发条件设置
- 可以工作在 RUN, SLEEP, STOP 多个模式
- 低功耗模式下 ADC、REF、Clock 供电自动控制，无需软件干预
- 支持两个 ADC 模块同时同步并行工作

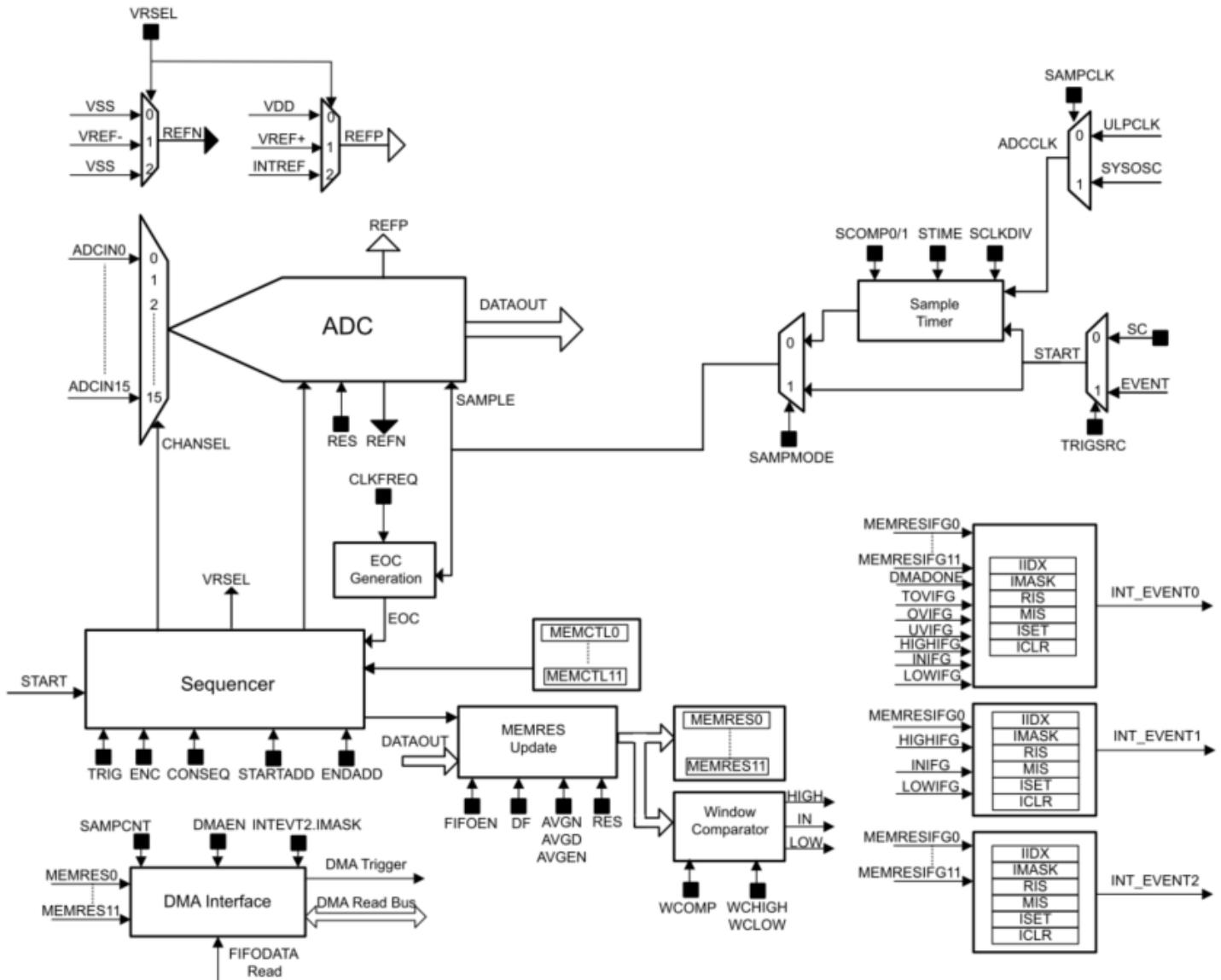


图 1-2. ADC 结构框图

1.2 MSPM0 DMA 概述

DMA (direct memory access) 提供了从一个地址到另一个地址搬移数据的功能，在此期间无需 CPU 干预，比如，DMA 可以将 ADC 的转换结果搬到 MCU 的内部 RAM。MSPM0 提供了多达 16 个 DMA 通道，这将会大大增强 CPU 内部的数据吞吐能力，同时也可以降低 CPU 负载，降低功耗，提高整个系统的效率。

MSPM0 DMA 特点:

- 提供多达 16 个 DMA 通道
 - DMA 通道可进行优先级编程
 - 支持 Byte (8-bit), short word (16-bit), word (32-bit) , long word (64-bit) 或者针对 mixed byte, word 的传输能力
 - 传输长度最大可以支持到 64k 个数据（单个数据的位宽根据配置决定）

- 支持 6 种不同的地址模式
- 支持 Single 和 Block 传输模式
- 支持特殊的 Fill Data 到 Block 的模式
- 支持特殊的 Data Table 传输到特定地址的模式

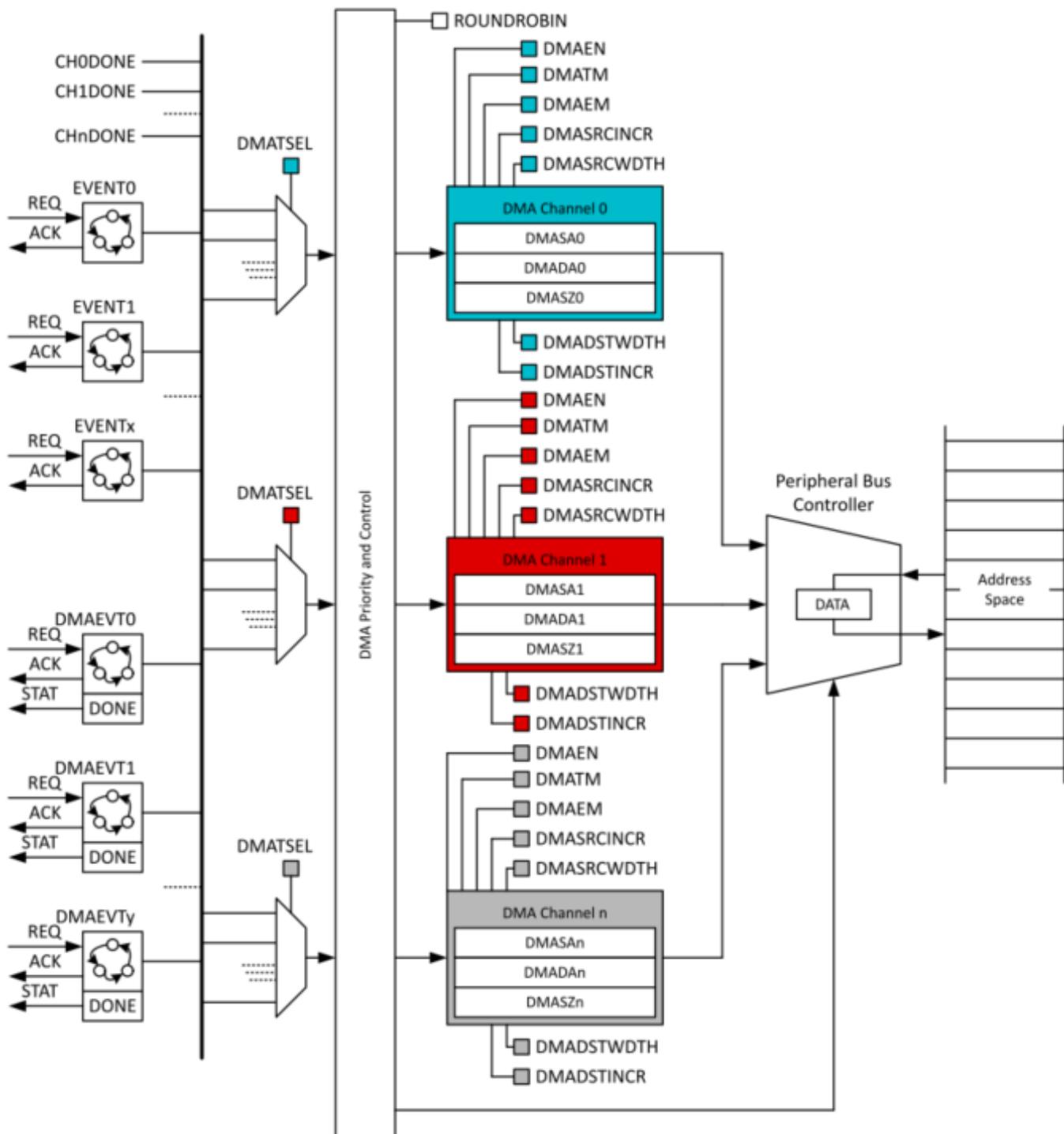


图 1-3. DMA 结构框图

1.3 MSPM0 Event 概述

MSPM0 最新集成的 Event Manager 功能提供了 peripheral-to-peripheral, peripheral-to-DMA, and peripheral-to-CPU (IRQ) 的 event 联通功能, 各个 event 的发送端 (publishers) 和接收端 (subscribers) 通过一系列的静态路由和可编程路由通道进行连接。

这些通过 Event Manager 进行传输的 Event 包含:

- 外设 Event 作为 IRQ 传输到 CPU (例如: RTC 中断传输到 CPU)
- 外设 Event 作为 DMA 触发事件传输到 DMA 模块 (例如: UART 数据接收触发 DMA 的传输请求)
- 外设 Event 传输到另一个外设去直接触发这个外设的动作 (例如: TIMx Timer 这个外设发送一个周期性的 Event 到 ADC 的事件接收端, 然后 ADC 使用这个 Event 去周期性的触发其开始采样这个动作)

如果为了正确的处理某个 Event 而不得不请求改变 Power and/or Clock 的配置, Event Manager 也提供了针对 PMCU (power management and clock unit) 的接口。例如, 一个外设发送一个 Event 到 DMA, 但是 MCU 当前处于 STOP/STANDBY 的低功耗状态, DMA 默认是关闭的, 这个时候 Event Manager 会与 PMCU 单元进行协商, 并开启对应的 DMA 通道, 这样就可以在低功耗情况下且无需 CPU 干预的情况下做到 Event 触发和执行。

1.3.1 事件路由结构 (Event Fabric Routing)

MSPM0 提供了三种不同的 Event 路由路径:

静态事件路由 (Static Event Route) : 静态路由是一种点对点的路由, 它提供了 publisher 和 subscriber 的直接路由, 这类路由主要用在 CPU 的中断连接上, 类似于通用 MCU 的中断管理。

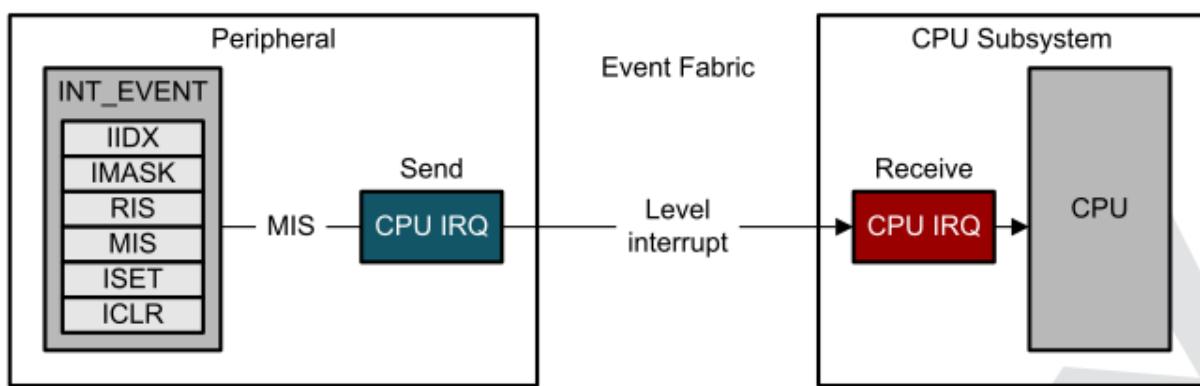


图 1-4. 静态事件路由 (Static Event Route)

DMA 事件路由 (DMA Event Route) : DMA 路由是一种外设到 DMA 的静态路由, 它有一些额外的信号去传输 DMA Done 状态到触发这个 DMA 的外设, 那些可以产生 DMA 触发源的外设具有一个指定的静态路由从外设到 DMA。

DMA 路由针对一些特定外设还包含了边带信号 (sideband signals) 去指示 DMA 传输队列已经完成。例如, DMA 可以在 UART TX 的 DMA 触发后去设置从 SRAM Buffer 到 UART TX 要传输 N Bytes, 根据 UART 的每一个 DMA 触发, DMA 会应答传输已经成功, 当到达 N Bytes 的时候, DMA 会发出一个边带结束信号 (sideband complete signal) 到 UART, 这时 UART 可以利用这个信号去发送 CPU 请求中断。

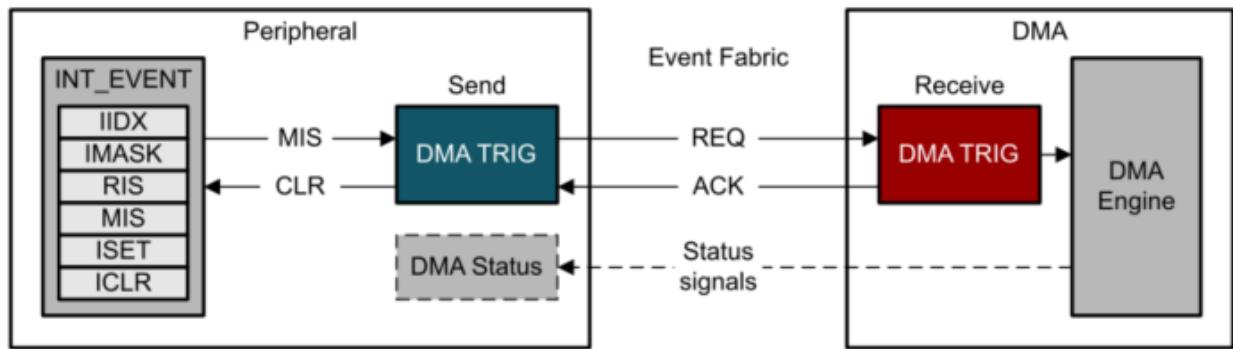


图 1-5. DMA 事件路由 (DMA Event Route)

通用事件路由 (Generic Event Route) : 通用事件路由可以进行 1:1 或者 1:2 的路由，外设可以利用其中一个通用路由路径去发送事件到另一个或者两个外设、DMA 或 CPU。

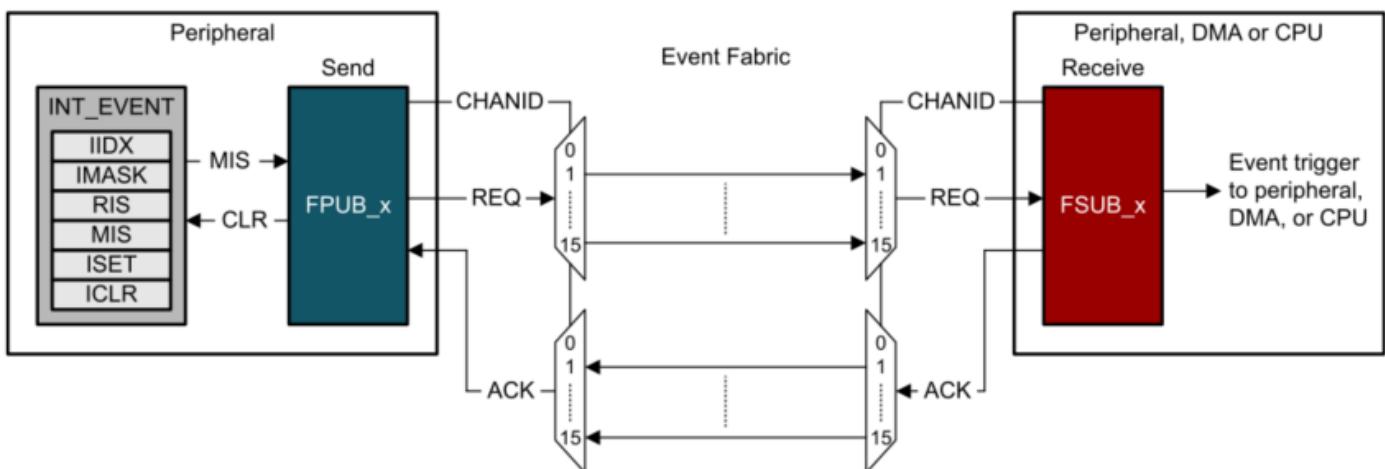


图 1-6. 通用事件路由 (Generic Event Route)

CHANID	Generic Route Channel Selection	Channel Type
0	No generic event channel selected	N/A
1	Generic event channel 1 selected	1 : 1
2	Generic event channel 2 selected	1 : 1
3	Generic event channel 3 selected	1 : 2 (splitter)

图 1-7. 通用事件路由 (Generic Event Route)

1.3.2 事件路由表 (Event Routing Map)

下图展示了 MSPM0 Event Manager 所涉及的事件路由表，我们可以看到，类似 UART、SPI 和 I2C 可以产生 CPU IRQ 静态路由事件，同时也可以产生 DMA 事件路由。类似 GPIO 和 ADC 可以产生 CPU IRQ 静态路由事件，也可以使用通用事件路由。例如，利用通用事件路由，通过连接 GPIO 的 FPUB_x 和 ADC FSUB_0 到同一个通用事件路由通道，就可以直接通过 GPIO 事件触发 ADC 采样。

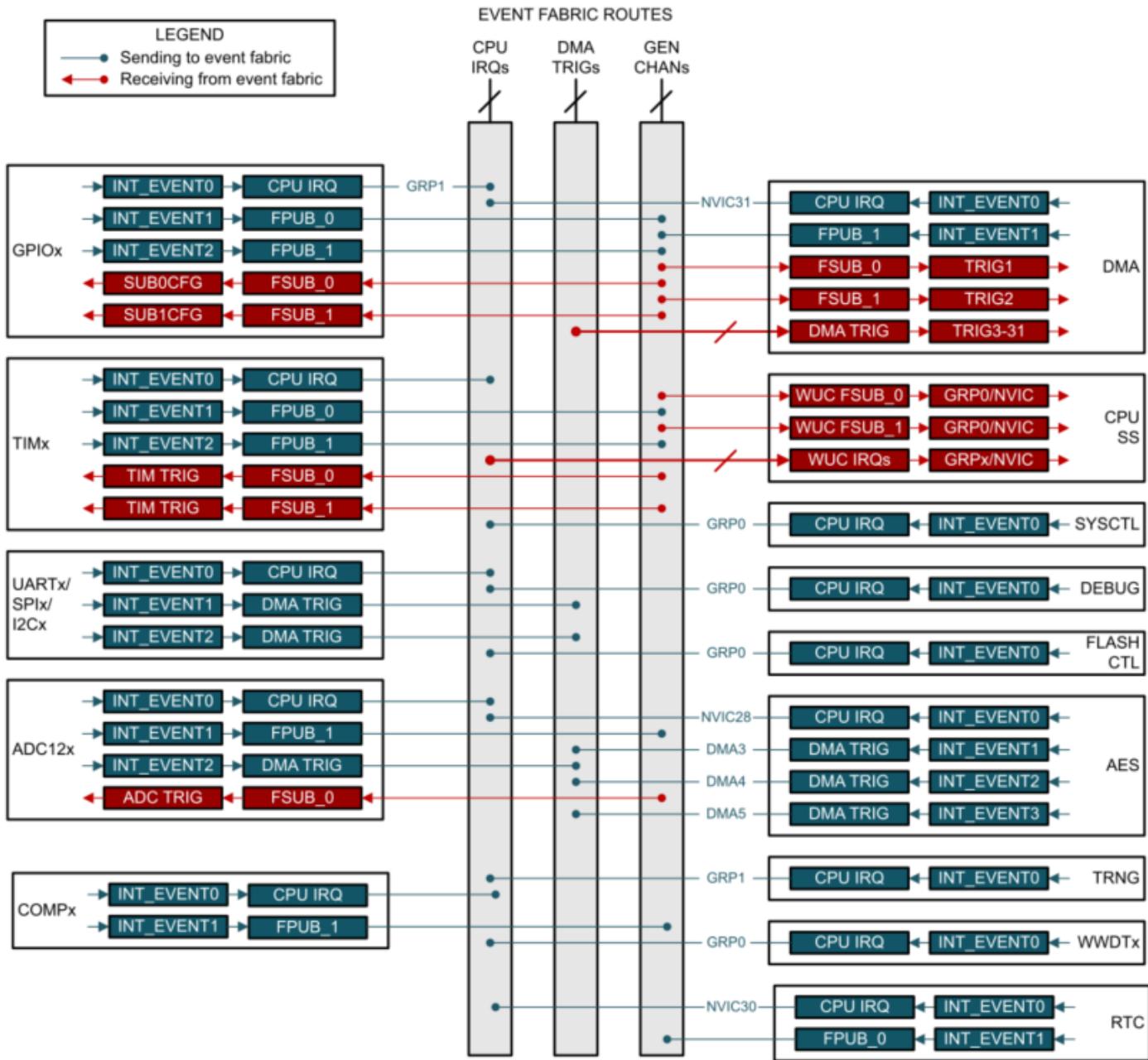


图 1-8. 事件路由表 (Event Routing Map)

2 使用 Timer 进行周期性 ADC 采样触发

2.1 MSPM0 Timer 概述

MSPM0 提供最多达 7 个 Timer 模块，分为 Timer G、Timer A 和 Timer H，具有多种增强功能，如 PWM 互补输出，死区控制，正交编码器输入功能，紧急刹车功能，高精度 24bit PWM 输出，触发 ADC、DAC 及 DMA 传输功能等，针对电机控制驱动等相关应用做了增强设计。

在本应用文档中，我们利用 Timer G 的周期性归零事件触发 ADC 采样，无需产生中断，无需 CPU 干预。

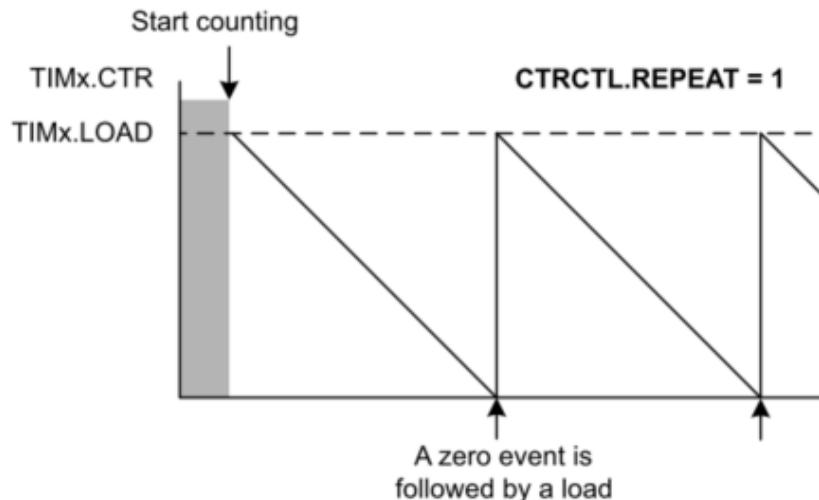


图 2-1. 周期性定时器 (Repeat Down Counting Mode)

在这种情况下，Timer G 作为 Event 的 Publisher 发送事件，ADC 作为 Subscriber 接收事件，通过 Event Manager 里面的通用事件路由通道，来建立 Timer G 到 ADC 采样的触发。

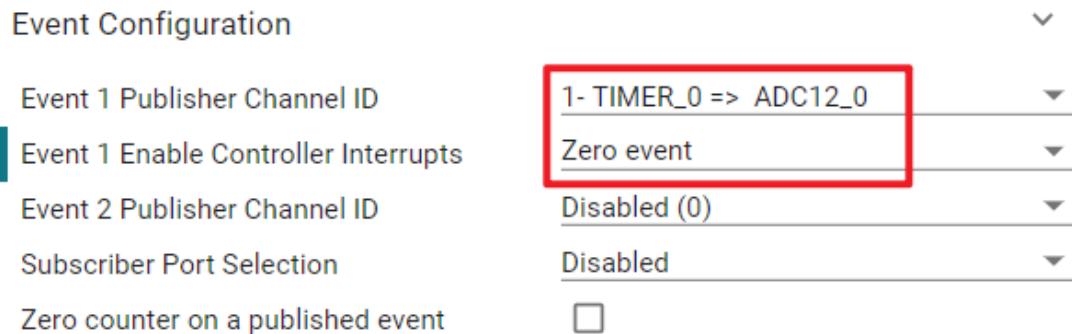


图 2-2. Timer G 事件配置



图 2-3. ADC 事件配置

3 利用 Event 驱动 DMA 进行 ADC 数据搬移

在 Timer 周期性触发 ADC 采样的情况下，每次触发，ADC 都会按照顺序对 8 个通道进行顺序采样，其采样的窗口可以通过独立的采样定时器来设定。

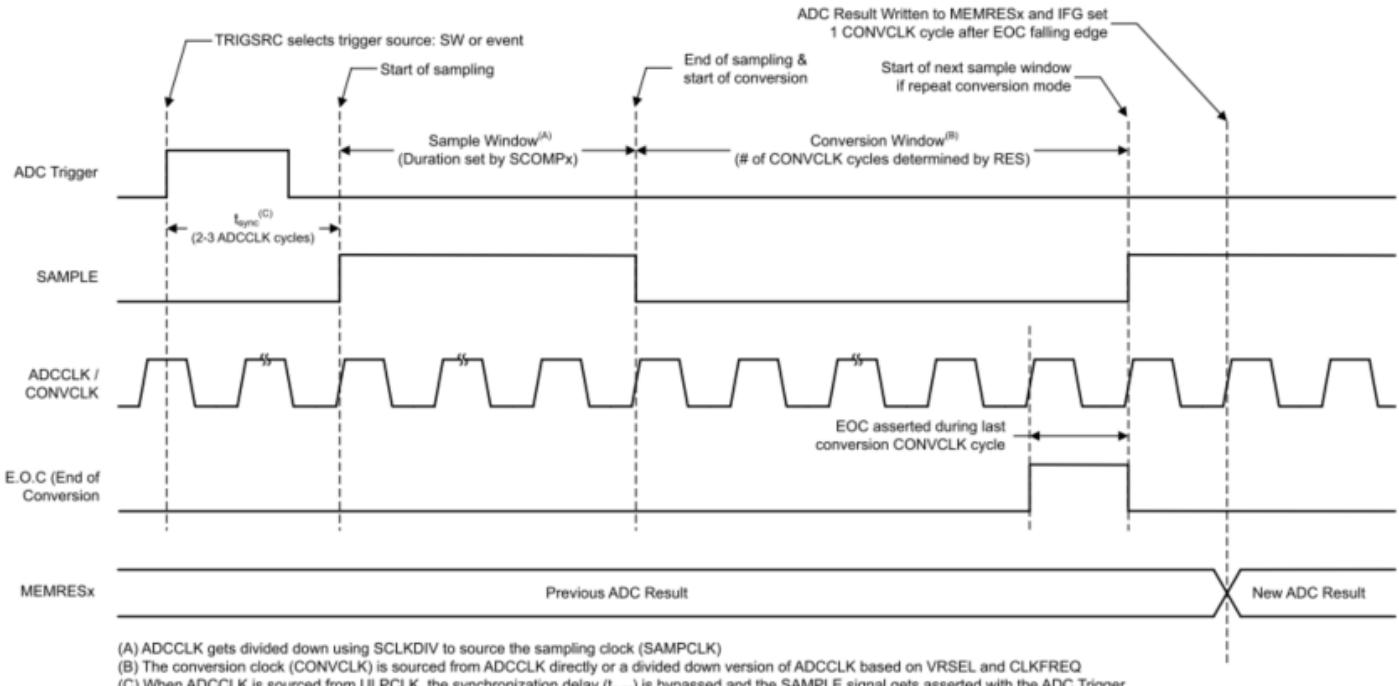


图 3-1. ADC Auto Sampling 模式

ADC Conversion Memory 0 Configuration

Name	0
Input Channel	Channel 0
Device Pin Name	PA27
Reference Voltage	VDDA
Sample Period Source	Sampling Timer 0

图 3-2. ADC 采样窗口配置

Advanced Configuration

Conversion Resolution	12-bits
Enable FIFO Mode	<input checked="" type="checkbox"/>
Power Down Mode	Manual
Desired Sample Time 0	1 us
Actual Sample Time 0	1.00 μ s
Desired Sample Time 1	10 us
Actual Sample Time 1	10.00 μ s

图 3-3. ADC 采样窗口 Timer 配置

当 ADC 转换结果输出的时候，会按照顺序将转换值放置在 **MEMRES0~MEMRES7** 这 8 个结果寄存器内，这时，我们就可以利用 **MEMRESx** 来触发 DMA 的传输。

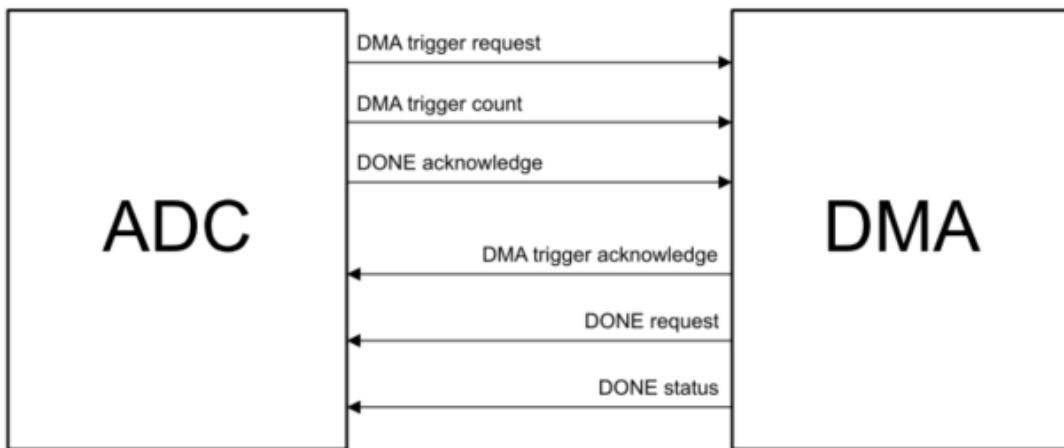


图 3-4. ADC DMA 内部接口

这里 **MSPM0** 在 **MEMRESx** 这里提供了两种方式来进行 DMA 的数据搬移，一种是不使用 **FIFO** 功能，一种是使用 **FIFO** 功能，这两者在 **DMA** 配置和传输模式上有较大的差别，需要根据应用本身的需求来进行选择。

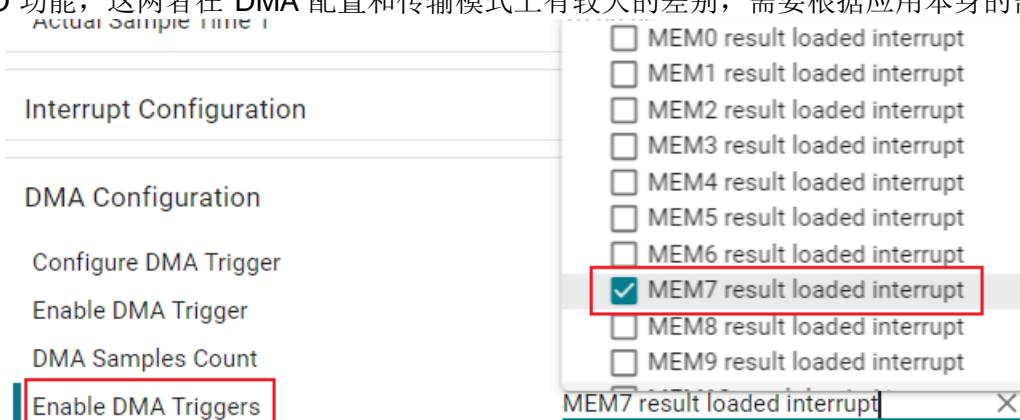


图 3-5. ADC DMA 触发源配置

在没有 **FIFO** 功能的情况下，整个 **DMA** 的触发和数据搬移比较简单直接，可以通过 **MEMRES7** 结果寄存器加载事件来触发 **ADC** 的 **DMA** 开始搬移数据，8 个通道的数据搬移结束后，可以触发 **ADC** 的 **DMA Done** 中断，这样就完成了一次搬移。这样的方式在本应用文档中要求的 8 通道周期采样是可以实现的，但是每 8 个通道数据搬移结束后会产生中断来打断 **CPU**，在中断应用程序内需要再次使能 **ADC Conversion** (**ADC** 在非 **Repeat** 模式下每次转换结束会自动清除 **ENC bit**) 和 **DMA Enable**，效率比较低，我们的目标是在持续采样 64 次 8 通道之后再触发中断，通知 **CPU** 对数据进行处理，这意味着 **DMA** 搬移了 512 个 **ADC** 结果值之后，再去响应中断。

4 使用 **FIFO** 以优化 **DMA** 的搬移效率

为了满足尽可能的避免 **CPU** 被中断，保证整个过程全部自动完成，我们需要启用 **ADC** 的 **FIFO** 功能来配合 **DMA** 进行数据搬移。

这里我们需要注意到，首先要将 **ADC** 的采样模式配置为 **Repeat Sampling** 模式，这意味着每次 8 通道采样完成后 **ADC** 会自动启动再次轮询这 8 个通道，但是这里也带来一个问题，我们需要 **ADC** 严格按照每 1ms

轮询采样一次，如果简单的使用 Repeat Sampling 模式，ADC 会持续不断的进行采样转换，所以我们要对 ADC 的轮询采样触发条件进行配置。

在 Repeat Sequence 模式下，如果在 MEMCTL 里配置了 TRIG 位，这意味着 ADC 的采样动作需要再一次被触发源触发，才能继续下一步的采样，所以我们需要在第 8 个通道的 MEMCTL7 这里对 TRIG 位进行设置，确保在 Repeat Sequence 模式下，当第 8 个通道的转换结束后，停止轮询到下一个通道（通道 0），等待下一个 Timer 的 1ms Event 来触发 ADC。

		Trigger Mode	
		SW Trigger	Event Trigger
Sampling Mode	AUTO	<ul style="list-style-type: none"> Set SC bit to start the sample phase (duration determined by sample timer) Conversion starts once sample phase is over In single channel single conversion, ENC is cleared when conversion is over SC bit is automatically cleared once the trigger is captured For repeat and sequence modes, if TRIG is set in MEMCTL, the SC bit needs to be set for the next conversion to proceed 	<ul style="list-style-type: none"> HW trigger starts the sample phase (duration determined by sample timer) Conversion starts once sample phase is over In single channel single conversion, ENC is cleared when conversion is over For repeat and sequence modes, ADC waits for HW trigger or automatically starts the next conversion based on TRIG setting
	MANUAL	<ul style="list-style-type: none"> Set SC bit to start the sample phase (SC bit is not automatically reset) Clear the SC bit to end the sample phase and start the conversion In single channel single conversion, ENC bit is cleared when conversion is over Repeated/sequential conversion modes are NOT supported in this configuration 	ADC operation is NOT supported in this configuration

图 4-1. ADC 采样模式和触发条件配置表

ADC Conversion Memory 7 Configuration

Name	7
Input Channel	Channel 7
Device Pin Name	PA22
Reference Voltage	VDDA
Sample Period Source	Sampling Timer 0
Optional Configuration	
Averaging Mode	<input type="checkbox"/>
Burn Out Current Source	<input type="checkbox"/>
Window Comparator Mode	<input type="checkbox"/>
Trigger Mode	Valid trigger will step to next memory conversio...

图 4-2. ADC 采样触发条件配置

由于使能了 FIFO，MEMRESx 寄存器不推荐直接去用 DMA 来访问，而是通过 FIFO 的地址作为 DMA 源地址来进行数据搬移。这里需要注意的是，当使能 FIFO 后，ADC 采样转换的结果会自动组合为 32Bit 位宽的数据存储在 FIFO 中，这样在配置 DMA 的时候，需要把源地址数据宽度配置为 4Bytes，目的地址数据也为 4Bytes。

由于要使用 DMA 来搬移数据，必须针对 FIFO 使能后的情况对触发 DMA 的条件进行特殊配置。在 MSPM0 的 ADC 模块内部，没有专门针对 FIFO 的深度进行配置的寄存器，而是用 MEMRESx 的位置来间接表示 FIFO 的深度，由于 8 通道在 12 个 MEMRESx 的结果寄存器中存储一次后只剩余了 4 个，接下来再次循环的话会存在高位 4 个通道和地位 4 个通道循环错位的情况（下一次采样转换结果的前 4 个会放在 MEMRES8、9、10、11，后 4 个会放在 MEMRES0、1、2、3），这样会导致 DMA 的触发条件无法设置（会导致搬移数据错位），因此我们采取了折中方式，将 DMA 触发的条件设置为 MEMRES3、7、11 来开始搬移数据，这意味着每 4 个结果就搬移一次，12 个结果寄存器刚好使用 3 次搬移完成，不存在剩余数据的情况，解决了 FIFO 触发 DMA 搬移的问题。

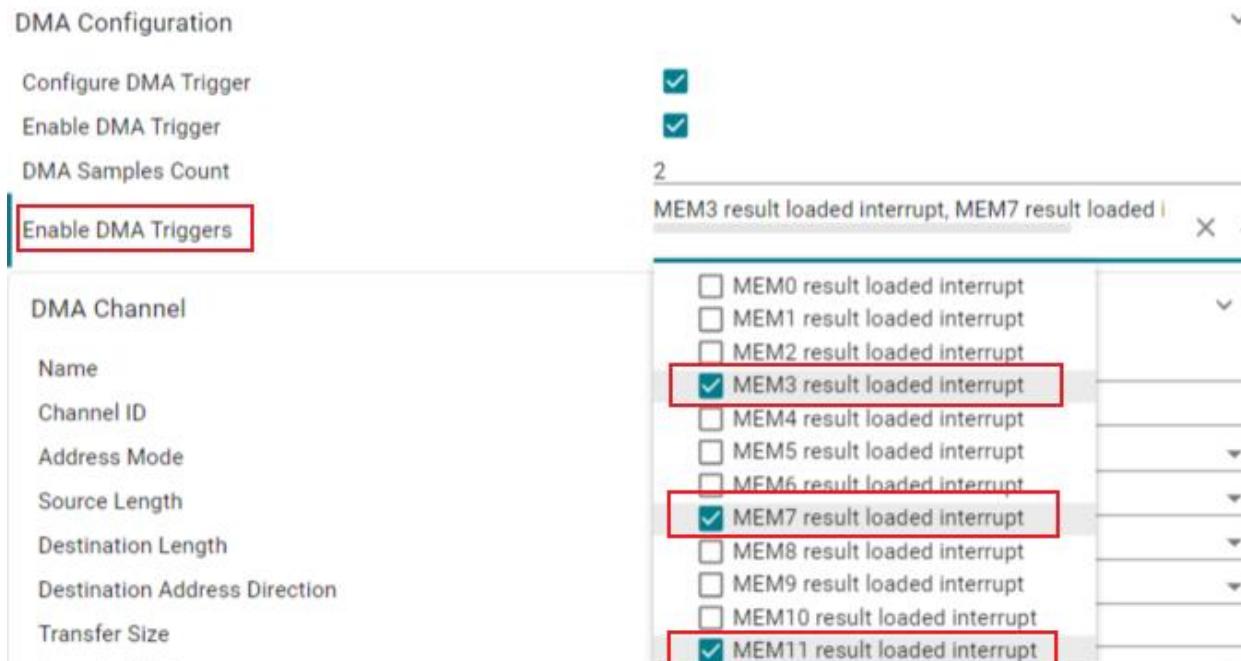


图 4-3. ADC FIFO 模式下 DMA 触发条件配置

5 使用双缓冲以留足时间余量给 CPU 进行数学运算处理

在本应用中，ADC 轮询采样得到的数据会缓存到 RAM 中，通过 CPU 进行后续的数据处理，一般情况下，为了保证 ADC 数据采样与 CPU 数据处理的可靠性和独立，建立双缓冲 Buffer 是必要的。

要实现 Buffer 的切换，我们需要对 DMA 传输的结果进行判断，这里利用 DMA 的传输长度来进行设置，在本应用中，8 通道一共采样 64 次，传输数据的长度为 32 次（注意：由于使用了 FIFO，每次搬移数据的宽度为 4Bytes，ADC 的转换结果为 2Bytes，所以 DMA 真正的传输次数为 $64 / 2 = 32$ 次）。在完成一次完整的 DMA 数据搬移后，会有 ADC 的 DMA Done 中断产生，这意味着一次完整的数据传输完成，可以进行 Buffer 切换，在操作 DMA Buffer 切换的过程中，需要注意这时的 DMA 通道已经自动变为禁止状态，对 DMA 的目标地址进行重新配置后，同时还需要重新配置传输长度，再次使能 DMA 通道，最后使能 ADC 的 DMA 功能，即可完成 Buffer 切换。

6 总结

在多路信号采样应用中，更灵活的使用 MSPM0 MCU 产品的 ADC、DMA 和 Event 机制，可以在几乎无需 CPU 干预的情况下，灵活的进行数据采样和传输，更进一步可以利用 ADC 模块中的硬件平均器来对数据做初步的滤波处理，提高了整个系统的运行处理效率，降低了不必要的中断响应，无论从功耗方面还是系统健壮性方面，都带来了不错的性能改进。

7 参考文献

1. [MSPM0G350x Mixed-Signal Microcontrollers Datasheet \(SLASF83\)](#)
2. [How Arm Cortex-M0+ MCUs optimize general-purpose processing, sensing and control](#)
3. [MSPM0 G 系列 MCU 硬件开发指南 \(Rev. A\) \(SLAAE76A\)](#)
4. [MSPM0 MCU 快速参考指南 \(Rev. A\) \(SLAAE70A\)](#)

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2023, 德州仪器 (TI) 公司