

## ADI公司数字加速度计中先进先出(FIFO)缓冲器的使用

作者: Christopher J. Fisher、Tomoaki Tsuzuki和James Lee

### 简介

当今许多产品对用户交互、手势识别和省电的要求越来越高,因此各种不同的应用纷纷采用惯性传感器。通过检测设备的运动状态,非活动期间可以关闭一些特性,以节省电能。惯性传感器能够区分敲击轴与敲击方向,这就为实现新颖而独特的用户输入方式创造了条件。

ADI公司提供多种多样的加速度计,包括单轴、双轴、三轴模拟和数字加速度计。超低功耗、三轴数字加速度计

[ADXL345](#)内置灵活的先进先出(FIFO)缓冲器。利用FIFO可进一步增强对用户手势的辨别能力,实现进一步省电,提高系统性能,同时降低对主处理器交互的需求。

本应用笔记以ADXL345为基础说明FIFO原理及其工作模式,从而讨论ADI公司数字加速度计所使用的FIFO技术。文中提供了各种模式的配置示例,并且讨论了FIFO用于信号处理和省电的一些例子。

## 目录

简介 .....	1	FIFO配置 .....	7
FIFO描述 .....	3	FIFO模式 .....	7
从数据寄存器中读取数据 .....	4	FIFO应用示例 .....	11
从FIFO中读取数据 .....	5	省电 .....	11
通信速度、数据速率和终止时间 .....	6	信号处理和滤波 .....	12
FIFO状态监控 .....	6		

## FIFO描述

FIFO最多能够保存32个样本数据集。每个样本数据集由一个x轴样本、一个y轴样本和一个z轴样本组成。一个x轴样本是指通常保存在DATA0和DATA1寄存器中的数据，y轴和z轴样本则是指保存在相应寄存器中的数据。此外，还有一个样本数据集保存在加速度计的输出滤波器中，形成FIFO的第33级。图1给出了FIFO的示意图。

加速度数据经过检测和数字化处理之后，输出滤波器释放样本。然后，此数据被置于FIFO中距数据寄存器最近的可

用位置。当读取数据寄存器时，将获得FIFO[0]中的数据，然后将从FIFO中移除该数据，使得堆栈中的其余数据可以向数据寄存器移近一级。也就是说，读取FIFO[0]之后，FIFO[1]中的样本移入FIFO[0]，FIFO[2]中的样本移入FIFO[1]，依此类推。读取之后，如果没有新样本可以移入FIFO[0]，则旧数据将保留，直到输出滤波器提供新数据并替代FIFO[0]中的样本。当FIFO填满时，输出滤波器的操作取决于工作模式。

OUTPUT FILTER	X-AXIS		Y-AXIS		Z-AXIS	
	DATA0	DATA1	DATA0	DATA1	DATA0	DATA1
FIFO[31]	DATA0[31]	DATA1[31]	DATA0[31]	DATA1[31]	DATA0[31]	DATA1[31]
FIFO[30]	DATA0[30]	DATA1[30]	DATA0[30]	DATA1[30]	DATA0[30]	DATA1[30]
⋮	⋮	⋮	⋮	⋮	⋮	⋮
FIFO[2]	DATA0[2]	DATA1[2]	DATA0[2]	DATA1[2]	DATA0[2]	DATA1[2]
FIFO[1]	DATA0[1]	DATA1[1]	DATA0[1]	DATA1[1]	DATA0[1]	DATA1[1]
FIFO[0]	DATA0[0]	DATA1[0]	DATA0[0]	DATA1[0]	DATA0[0]	DATA1[0]
DATA REGISTER (0x32 TO 0x37)						

图1. FIFO缓冲器示意图

00234-001

## 从数据寄存器中读取数据

建议对数据寄存器的所有读操作都应当用多字节读操作来完成。多字节读操作是指直到获得多字节读取中的最后一个字节才结束通信的读操作。这种读操作可以确保从寄存器读取的数据与同一样本相对应。如果执行单字节读操作，则数据在两次读操作之间可能会改变，导致所读取的数据混杂着不同样本的数据。

对于SPI三线式或四线式通信，通过设置多字节位并连续发送各字节的8时钟脉冲集，可以实现多字节读操作。设置多字节位后，每经过一个8时钟脉冲集，寄存器指针便

移向下一个地址，如图2所示。有关多字节位的位置说明，请参考ADXL345数据手册。

对于I2C通信，多字节读操作的执行方式与SPI通信相似，但要注意执行读操作所必需的I2C协议。启动读操作后，连续发送9时钟脉冲集(I2C通信中，第9个时钟脉冲用于应答位)可将寄存器指针移向下一个地址。有关I2C通信中执行读操作的更多信息，请参考ADXL345数据手册或《UM10204 I2C总线规范和用户手册》03版(2007年6月19日，NXP Semiconductors提供)。

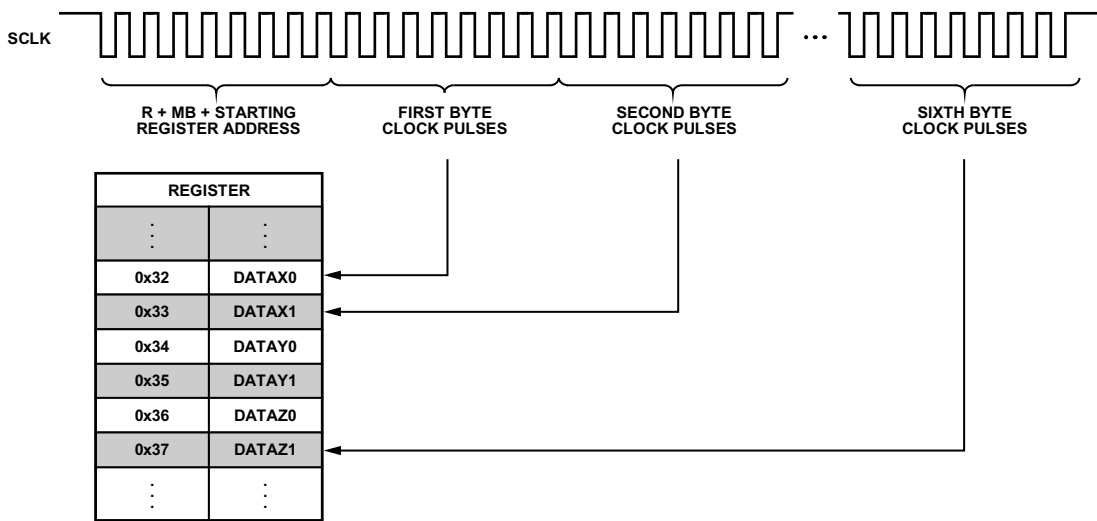


图2. ADXL345中多字节SPI读操作期间的寄存器指针移位

08234-002

### 从FIFO中读取数据

当使用FIFO时，读取数据寄存器将返回FIFO[0]中存储的样本。要启动移除旧FIFO[0]值并下移样本的过程(也称为弹出FIFO)，必须收到DATAZ1寄存器的最后一个时钟脉冲，否则通信将终止。SPI传输期间，解除CS引脚有效可以终止通信。主器件通过发出停止命令可终止I<sup>2</sup>C通信。要读取下一个样本，请再次执行多字节读操作，包括必要的寄存器寻址，从初始数据寄存器开始，如图3所示。

与从数据寄存器中读取数据相似，如果从FIFO中读取每样本一个字节以上的数据，则需要执行多字节读操作。使用FIFO时执行单字节读操作将导致数据在两次读操作之间移位，从而混杂来自不同数据集的样本，因为一个读操作结束将导致FIFO弹出。例如，如果使用一个单字节读操作读取x轴的LSB，然后使用另一个单字节读操作读取x轴的MSB，则将导致FIFO在两次读取之间弹出，从而返回相邻样本中的字节，而不是同一样本中的字节(见图4)。

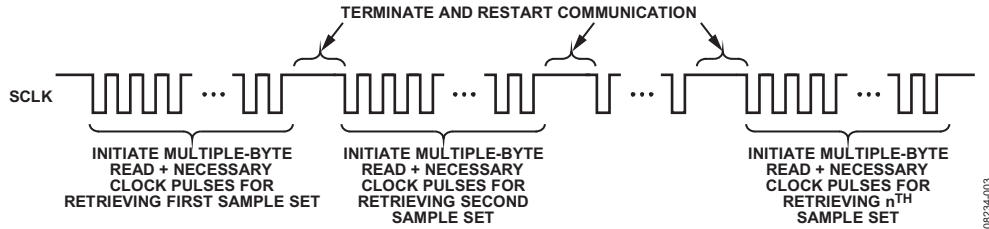


图3. 两次FIFO读操作之间的通信终止和再启动将导致FIFO弹出

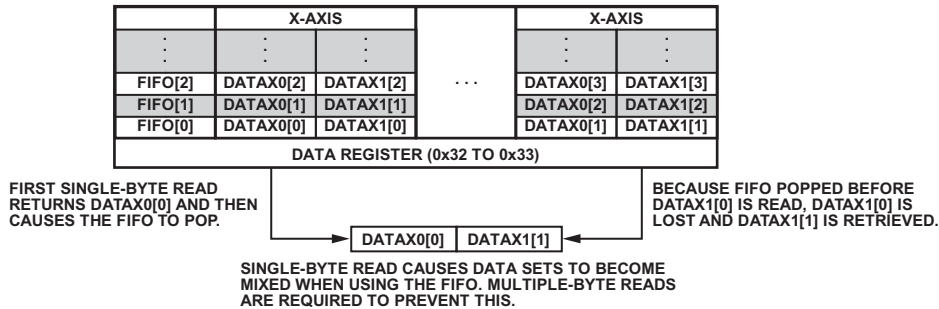


图4. 使用FIFO时的单字节读操作导致样本混杂

## 通信速度、数据速率和终止时间

从FIFO中读取数据的速率取决于通信速度，因此必须选择适当的数据速率，使得数据填充FIFO的速率不慢于从FIFO中读取数据的速率。I<sup>2</sup>C协议需要开始位、从地址加写标识、寄存器地址、重新启动、从地址加读标识才能开始多字节读操作。对于快速模式I<sup>2</sup>C，仅为开始多字节读操作所需的最少时间约为70 μs。如果要读取所有6字节数据，则获得一个FIFO样本所需的最少时间约为207 μs，其中包括所有最少启动和停止设置与保持时间以及最少周期时间。基于此，当以最大数据速率3200 Hz(相当于312.5 μs的周期时间)工作时，从FIFO中读取数据的速率仍然可以快于填充FIFO的速率，但如果使用较慢的I<sup>2</sup>C通信速度，则可能需要调整数据速率。I<sup>2</sup>C通信中，主机控制器的效率对可以使用的最大数据速率也有影响，确定此值时应予以考虑。

SPI开始读操作则没有同样的要求，因此以相同的数据速率工作时，使用低于I<sup>2</sup>C的速度也能成功读取数据。系统设计人员应当验证，所选数据速率对于总线速度是合适的。

连续样本读取之间所需的时间(图3所示的通信终止和重新启动时间)取决于通信方式和速度。之所以有此要求，是因为读取两个样本之间必须弹出FIFO，即读取DATAZ1结束或解除CS引脚有效与开始读取下一个样本集之间至少需要约5 μs的时间。对于I<sup>2</sup>C，快速模式下开始读操作所需的最少时间(70 μs)完全能够满足这一5 μs要求。

对于5 MHz SPI通信，开始只读操作需要约1.6 μs，因此CS引脚应保持解除有效状态至少3.4 μs。类似地，对于1.6 MHz SPI通信，开始读操作至少需要5 μs，这说明：如果SPI速度低于1.6 MHz，则只须满足CS引脚保持解除有效状态的最短时间要求 $t_{CS,DIS}$ 。

## FIFO状态监控

FIFO\_STATUS寄存器用于监控FIFO的状态。FIFO\_TRIG位在“触发器模式”部分中讨论。条目数位表示FIFO中当前有多少样本。虽然条目数有6位长，但最大可能值为0x20或32(十进制)。请注意，如果在FIFO得以完全弹出之前读取FIFO\_STATUS寄存器，则条目数中的值将是不正确的，因为它只有在FIFO完成移位之后才会更新。如果执行多字节读操作，从数据寄存器开始，并意图在读取FIFO\_STATUS寄存器之后结束，则完成读取DATAZ1与开始读取FIFO\_STATUS之间至少需要间隔5 μs，以允许FIFO弹出。这可以通过时钟延时或停止时钟至少5 μs来实现，然后继续通信。

表1. FIFO\_STATUS寄存器 (只读)

D7	D6	D5	D4	D3	D2	D1	D0
FIFO_TRIG	X <sup>1</sup>	条目数					

<sup>1</sup>X = 无关

## FIFO配置

FIFO通过FIFO\_CTL寄存器进行配置。FIFO的工作模式由FIFO\_CTL寄存器中的FIFO\_MODE位决定和配置。触发器模式的触发器中断和样本数位也是在FIFO\_CTL寄存器中进行配置，其工作方式因所选的模式不同而异。

**表2. FIFO\_CTL寄存器（读/写）**

D7	D6	D5	D4	D3	D2	D1	D0
FIFO_MODE		触发器	样本数				

为了防止产生虚假中断或意外填充FIFO，应在使能相应的中断以及将器件置于测量模式之前，配置FIFO以及触发器模式的触发中断或水印。一般应当按照以下顺序配置数字加速度计：

1. 设置数据速率、测量范围、数据格式和偏移调整等数据参数。
2. 配置中断(勿使能)：阈值和定时值，并将中断映射至引脚。
3. 配置FIFO(如在使用)：模式、触发器中断(如果使用触发器模式)以及样本数位。
4. 使能中断：INT\_ENABLE寄存器。
5. 将器件置于测量模式：POWER\_CTL寄存器。

按照上述顺序进行配置可以防止FIFO为空且样本数位配置为默认值0时产生虚假水印中断，以及利用默认值使能中断而导致虚假中断。

### FIFO模式

FIFO支持四种工作模式：旁路模式、FIFO模式、流模式和

触发器模式。通过FIFO\_CTL寄存器中的FIFO\_MODE位可以配置各种模式。关于何值对应何种模式，请参考ADXL345数据手册。

### 旁路模式

旁路模式下，FIFO禁用。每次输出滤波器有新样本可提供时，新数据立即替代数据寄存器或FIFO[0]中存储的值。如果在新数据就绪之前再次读取数据寄存器，旧数据将保留到新样本可用时为止，然后新数据立即替代FIFO[0]或数据寄存器中的旧数据。如果正在读取当前数据样本时新数据可用，则双缓冲输出允许读操作先行完成，然后将新数据置于输出滤波器，防止读操作引起数据丢失。

这种模式下，DATA\_READY中断一般用于告知主器件：新数据已就绪，可供读取。DATA\_READY中断的配置方法如下：将INT\_MAP寄存器中的DATA\_READY位配置为所选的中断引脚，然后设置INT\_ENABLE寄存器中的DATA\_READY位。当新数据可用时，DATA\_READY所映射的中断触发；读取数据寄存器中的数据之后，该中断清除。DATA\_READY中断的默认极性为高电平有效。设置DATA\_FORMAT寄存器中的INT\_INVERT位，可以将DATA\_READY和所有其它中断配置为低电平有效。

**表3. INT\_MAP/INT\_ENABLE寄存器（读/写）**

D7	D6	D5	D4
数据更新	单击	双击	活动
D3	D2	D1	D0
非活动	自由落体	水印	溢出

### 通信速度、数据速率和终止时间

FIFO模式下，FIFO收集新样本，直到填满为止，然后丢弃新样本，直到FIFO中有空间可用。当通过读取样本而获得可用空间时，输出滤波器提供的最新数据再次开始填充FIFO。如果FIFO已填满，而样本读取速度不够快，则在填充FIFO时存储于其中的样本与有空间可用后移入的新样本之间会有样本被丢弃，导致数据脱节。这种现象如图5所示。FIFO模式下的正常工作方式是禁用DATA\_READY(如已使能)，而使用水印中断。FIFO模式下，当FIFO中存储的样本数(对应于FIFO\_STATUS寄存器条目数中的值)等于写入FIFO\_CTL寄存器的样本数位的值时，就会触发水印中断。使用水印中断，而不是DATA\_READY中断，允许FIFO填充至所需的级(存储在样本数位中)，然后执行几个连续的多字节读操作以清空FIFO，从而减少加速度计给主处理器造成的负担。要配置水印中断，应将一个非零值写入样本数位。所用值是一个系统级选择，取决于中断发生时中断的处理速度。如果中断处理很快，并且读取FIFO可以在新样本就绪之前开始，则可以选择一个接近32的值。如果中断发生与处理之间存在延迟，则应选择较低的值，以防FIFO丢弃样本。配置样本数位中的值后，通过INT\_MAP寄存器将水印中断映射至INT1或INT2，然后设

置INT\_ENABLE寄存器中的水印位，以使能水印中断。与DATA\_READY中断相似，水印中断将保持置位，直到出现导致中断消除的条件。这意味着，只要条目数中的值大于或等于样本数位中的值，水印中断就会保持触发状态。若要清除水印中断，应读取FIFO(通过数据寄存器)，直到FIFO中的样本数小于样本数位中存储的值。不过，为了防止水印中断频繁触发，致使FIFO的好处荡然无存，建议读取比最小数目多的数据样本。

FIFO在FIFO模式下利用水印中断工作的例子如图6所示。本例中，样本数位的值配置为6；因此，将一个样本压入FIFO[5]时，水印中断就会触发。再经过几个样本之后，中断得到处理，并且从FIFO中读取6个样本，使FIFO中的条目数小于6，这样就会清除水印中断。

请注意，使器件从其它FIFO工作模式进入旁路模式时，将导致FIFO被清空。因此，应在器件进入旁路模式之前，读取FIFO中的数据。器件从测量模式进入待机模式可保存FIFO的内容，但不会再有新样本填充FIFO。这在允许FIFO部分填充，并且FIFO没有在返回测量模式之前清除其中内容就进入待机模式时，可能会造成数据样本脱节。

OUTPUT FILTER	FIFO CONTENTS							
	DATA[0]	DATA[1]	...	DATA[31]	DATA[32]	DATA[33]	DATA[34]	DATA[35]
FIFO[31]	X	X	...	DATA[31]	DATA[31]	DATA[31]	DATA[31]	DATA[35]
FIFO[30]	X	X	...	DATA[30]	DATA[30]	DATA[30]	DATA[30]	DATA[34]
FIFO[29]	X	X	...	DATA[29]	DATA[29]	DATA[29]	DATA[29]	DATA[31]
FIFO[28]	X	X	...	DATA[28]	DATA[28]	DATA[28]	DATA[28]	DATA[30]
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
FIFO[2]	X	X	...	DATA[2]	DATA[2]	DATA[2]	DATA[2]	DATA[4]
FIFO[1]	X	DATA[1]	...	DATA[1]	DATA[1]	DATA[1]	DATA[1]	DATA[3]
FIFO[0]	DATA[0]	DATA[0]	...	DATA[0]	DATA[0]	DATA[0]	DATA[0]	DATA[2]

TIME →

MULTIPLE-BYTE READ OF 2 SAMPLE SETS, DATA[0] AND DATA[1], CAUSES FIFO TO SHIFT. DATA[32] AND DATA[33] ARE LOST BECAUSE ROOM WAS NOT AVAILABLE UNTIL AFTER DATA[34] WAS SAMPLED. BECAUSE DATA[34] IS STILL THE NEWEST DATA WHEN SPACE IS MADE AVAILABLE IN FIFO, IT IS SHIFTED INTO FIFO[30] BEFORE DATA[35] IS MADE AVAILABLE.

06234-005

图5. FIFO模式下的FIFO填充顺序，数据由于溢出而丢失



**流模式**

当FIFO以流模式工作时，FIFO持续不断地收集新数据样本。如果FIFO已满，则FIFO[0]中的最旧样本将被丢弃，其余样本下移，新样本则被压入FIFO[31]。最旧样本不断被丢弃，直到读取FIFO，新样本有空间可用为止。这一过程如图7所示。

流模式适合利用外部激励(而不是来自加速度计的激励)决定何时读取加速度数据的应用。外部激励可以是一个按钮操作，或是主处理器在某一时间要求加速度计提供信息。正因如此，流模式下一般不使用中断。不过，水印中断在流模式下的工作方式与在FIFO模式下相同，当FIFO中的样本数等于样本数位中存储的值时触发。

与流模式关系最密切的特性是INT\_SOURCE寄存器中的溢出位。当FIFO已填满，必须丢弃一个样本才能容纳新样本时，溢出位就会置1，表示数据由于FIFO已满而丢失。这在FIFO的所有工作模式下均相同，但在FIFO模式和触发器模式下，通常会在FIFO填满之前读取数据，以防数据丢

失。图7中，溢出位在输出滤波器提供DATA[32]时置1，因为DATA[0]已被丢弃。读取FIFO之后，该位清零，样本下移，FIFO中又有空间可用。这可以防止丢失更多数据，直到FIFO再次填满。

**表4. INT\_SOURCE寄存器 (只读)**

<b>D7</b> 数据更新	<b>D6</b> 单击	<b>D5</b> 双击	<b>D4</b> 活动
<b>D3</b> 非活动	<b>D2</b> 自由落体	<b>D1</b> 水印	<b>D0</b> 溢出

**触发器模式**

触发器模式下，FIFO以与流模式相似的方式开始工作，收集样本，直到填满为止，然后丢弃最旧样本，以为新样本提供空间。发生触发事件之后，总数等于样本数位中存储的值的最新样本得以保留，其余的最旧样本则被丢弃。然后，FIFO开始以与FIFO模式相似的方式工作，收集新样本，直到填满为止，然后丢弃新样本，直到FIFO中有空间可用。这一行为如图8所示，样本数值为15。

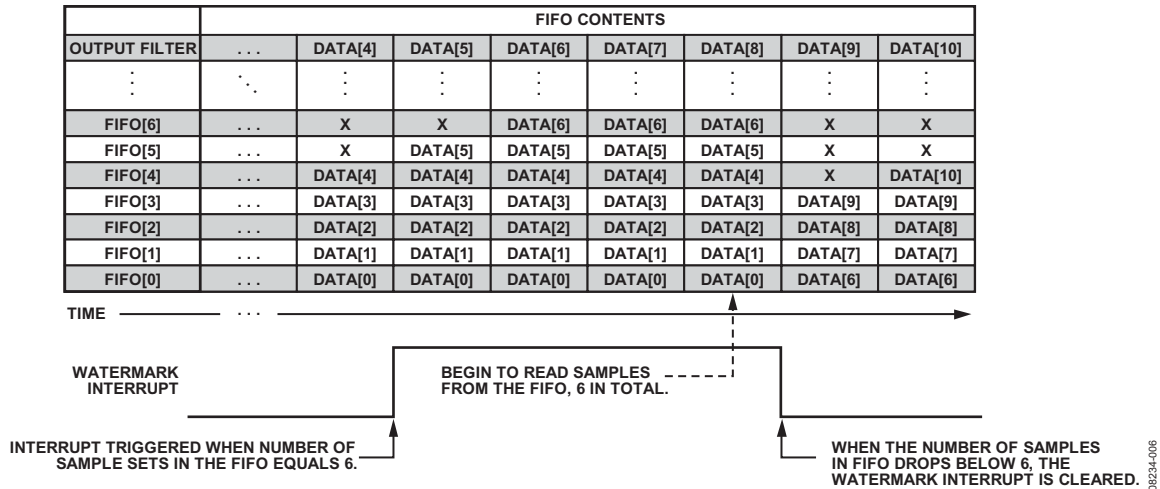


图6. FIFO模式工作示例，使用水印中断，样本数值为6

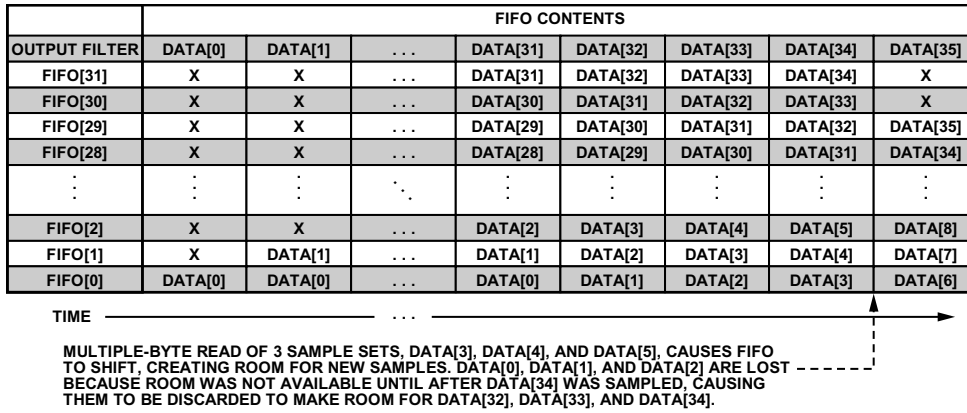


图7. FIFO以流模式工作，丢弃最旧样本

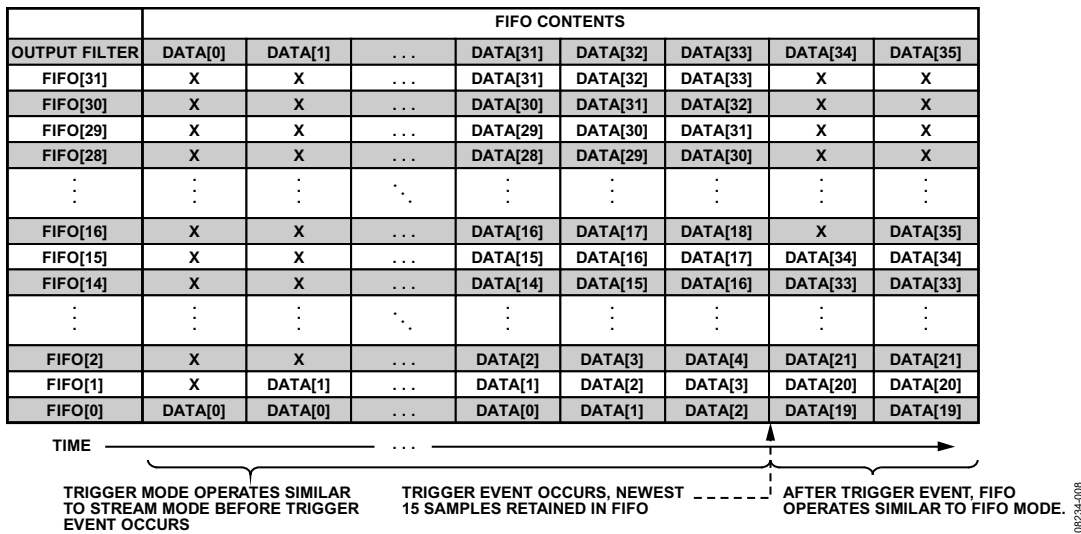


图8. FIFO以触发器模式工作，样本数值为15

触发器模式的触发事件是指加速度计固有的任何中断。如果选择了多个中断，则所发生的第一个中断充当触发事件。首先配置目标中断(参考ADXL345数据手册)，然后将其映射至INT\_MAP寄存器中的相应中断，然后在INT\_ENABLE寄存器中使能中断。配置中断后，配置FIFO\_CTL寄存器中的触发器位，使之对应于触发FIFO的中断。对于ADXL345，值0将选择INT1，值1将选择INT2。如果出现一个事件，导致触发器位所选的中断发生，则FIFO将保留样本数位中的值所确定的最新样本，然后在流模式与FIFO模式之间转换。建议不要将DATA\_READY、水印和溢出中断用作触发器模式的触发事件。

触发中断发生之后，读取FIFO直到清空为止。如果FIFO继续以FIFO模式工作，则可以调整样本数位中的值，以将主处理器从加速度计中获取数据的次数和配置水印中断的次

数降至最少，如“FIFO模式”部分所述。或者，也可以从一开始就配置水印中断，将其映射至另一中断引脚，并将主处理器配置为忽略该中断，直到发生触发事件后才第一次清空FIFO。然后可以调整样本数位中的值，并配置主处理器对水印中断做出响应。

由于触发事件之后必须调整FIFO中的数据，因此触发事件发生与开始读取FIFO之间至少需要5 μs的时间。经过5 μs后，FIFO适当地丢弃最旧样本，并移动所保留的样本。

若要复位触发器模式，以便识别其它触发器事件，首先应完全读取FIFO(如果需要数据)，然后通过FIFO\_CTL寄存器将器件置于旁路模式，以便丢弃FIFO中的所有剩余数据，并复位触发器。再配置FIFO\_CTL寄存器中的相应位，使器件返回触发器模式。最后，配置器件对下一个触发事件做出响应。

## FIFO应用示例

### 省电

除了可以减少主处理器与加速度计的通信次数外，FIFO还可以在器件等待惯性事件发生或记录数据的应用中实现省电。监控货运包裹的震动或计步器就是这种应用的例子；对于后者，输出并不需要每步更新一次，而是可以每隔几步更新一次。

鉴于ADI公司数字加速度计的功耗已经非常之低，因此不用时将主处理器和其它外设置于睡眠状态将可实现最大程度的省电。利用FIFO可以持续不断地收集加速度数据，而主处理器仅在FIFO接近填满时才通过水印中断唤醒，接近填满与否由样本数位中的值决定。其余时间里，当FIFO在收集数据且不需要处理器时，处理器可以进入睡眠状态，

从而显著降低系统功耗。此外，如果存在其它外设，也可以关闭这些外设，直到系统需要这些外设时才开启。

为了显示数字加速度计的FIFO的省电能力，图9给出了数据收集与功耗的典型时间图。ADXL345的功耗是在100 Hz输出数据速率下测得的，其电流和唤醒时间曲线与ADI公司的ARM7系列微控制器相当。当FIFO中有30个样本可用时，FIFO产生一个水印中断；数据通过SPI以5 MHz速率读取，读取30个完整样本约需500  $\mu$ s。

在图9所示例子中，唤醒主处理器，读取30个数据样本，然后返回睡眠模式约需3.5 ms。使用FIFO的平均功耗仅为460  $\mu$ A，而主处理器一直开启时，平均功耗超过11 mA。相比之下，省电接近96%。

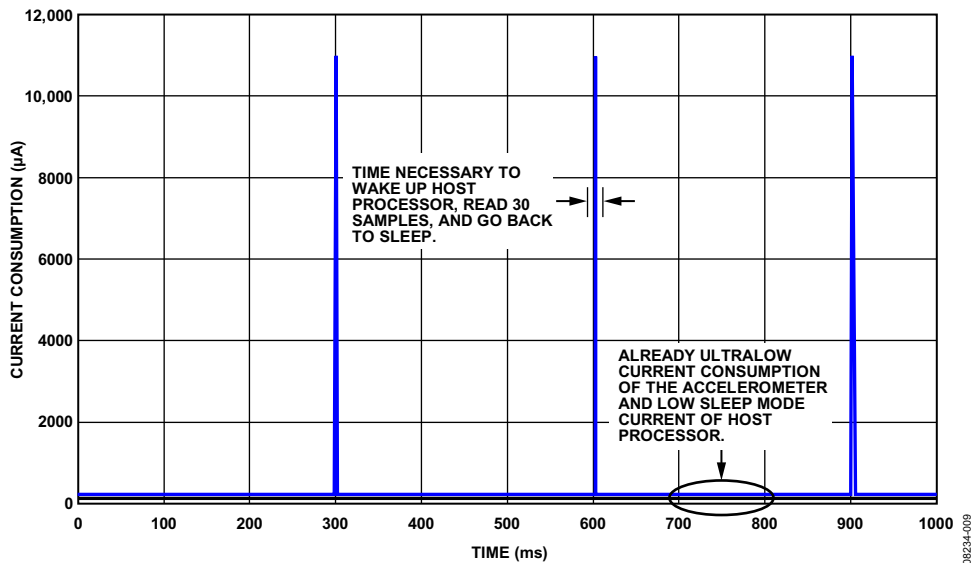


图9. 使用FIFO时将主处理器置于睡眠模式以实现省电

## 信号处理和滤波

一些应用中，例如倾斜检测，可能需要降低噪声和提高分辨率。由于ADI公司目前的数字加速度计产品的特性原因，以低于100 Hz的输出数据速率工作时，噪声性能并不优于100 Hz输出数据速率时的噪声性能。如果需要降低噪声，则应对输出进行外部滤波。3.9 mg/LSB比例因子大约相当于0.25°的倾斜分辨率，这足以满足大多数应用的需要。

如果需要更低的噪声或更高的分辨率，则可以运用滤波或过采样技术。无论是何种情况，均必须处理多个数据样本。这些应用允许FIFO填充到达到所需的样本数为止，主处理器无需一边不断地读取和存储数据，一边等待足够的

数据以便处理。一旦FIFO含有全部所需的数据，就可以从FIFO快速获取所有样本并予以处理，从而减少主处理器的负担。

对于简单的均值滤波器，噪声降低程度约等于求均值的样本数的平方根。例如，如果采集100个样本，其均方根(rms)噪声为1 LSB，则使用简单的四样本均值滤波器可将噪声降至约0.5 LSB rms。为证明这一点，可以使用ADXL345以100 Hz的输出数据速率采集100个数据点，然后将每四个样本平均为一个样本；当每组四样本就绪时，利用FIFO触发。这将获得明显的降噪效果，有效输出数据速率为25 Hz，因为主处理器只需每四个样本从器件读取一次数据，如图10所示。

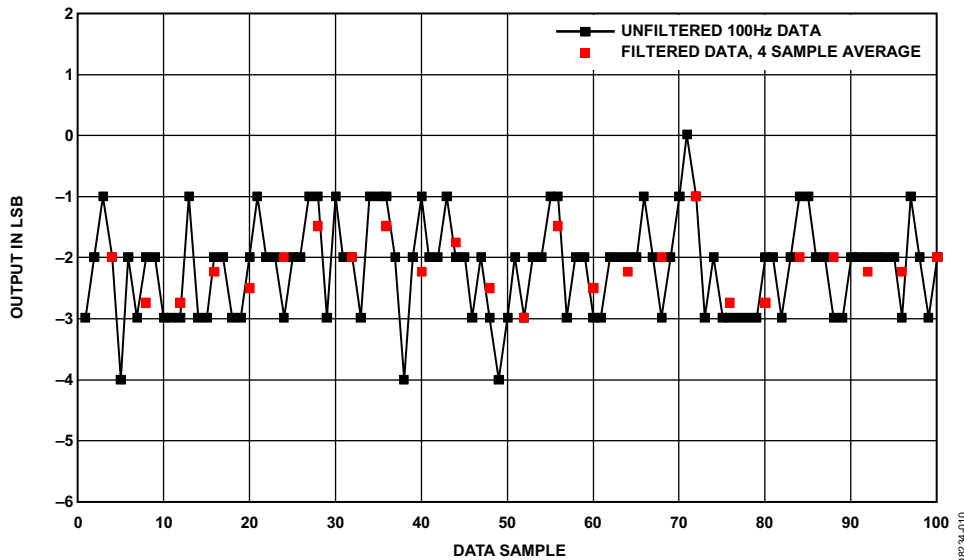


图10. 利用四样本均值滤波器和FIFO降低噪声