アナログ・デバイセズのデジタル加速度センサーに内蔵された先入れ、先出し(FIFO)バッファの利用

Christopher J. Fisher、Tomoaki Tsuzuki、James Lee 共著

はじめに

最新型の製品の多くでユーザーインターフェース、動作認識や低消費電力に対する要求が高まっている中で、幅広いアプリケーションに慣性センサーの採用が急速に進んでいます。例えば、加速度センサーを使い製品の動きをセンシングして特定の機能をオフにすることで、製品に動きがない間の消費電力を節減することができます。また、タップの振動を検出して入力軸とその方向を識別することにより、ユーザーに、新しいユーザーインターフェースを提供することが可能になります。

アナログ・デバイセズでは、単軸、2 軸、3 軸タイプのアナログおよびデジタルを含めた広範囲の各種加速度センサーを提供しています。超低消費電力、3 軸デジタル加速度センサーのADXL345 は、柔軟性に優れた FIFO バッファを内蔵しています。この FIFO を利用することで、アプリケーションの精度を向上させることができ、さらなる低消費電力化によってシステム性能が高まると同時に、加速度センサーからの割込みを減らすことでホスト・プロセッサの負担を軽減させることができます。

このアプリケーション・ノートでは、アナログ・デバイセズのデジタル加速度センサーについて解説し、ADXL345 を中心に FIFO とその動作モードを説明します。各モードの構成例を紹介するとともに、FIFO を利用した信号処理と低消費電力化の方法の例についてもいくつか説明します。

Rev. 0

アナログ・デバイセズ株式会社

本　　社／〒105-6891　東京都港区海岸 1-16-1　ニューピア竹芝サウスタワービル
電話 03（5402）8200
大阪営業所／〒532-0003　大阪府大阪市淀川区宮原 3-5-36　新大阪トラストタワー
電話 06（6350）6868

目次

FIFOの説明

FIFO は、最大で 32 のデータ・サンプル・セットを保持することが可能です。各データ・サンプル・セットは、x 軸サンプル、y 軸サンプル、z 軸サンプルで構成されます。x 軸サンプルは、通常 DATAX0 と DATAX1 の各レジスタに保持されるデータであり、y 軸サンプルと z 軸サンプルはそれぞれ該当するレジスタに保持されるデータです。これに加えて、加速度センサーの出力フィルタにも同様にもう 1 つのデータ・サンプル・セットが保持され、FIFO と合わせて合計 33 のデータ・サンプル・セットを保持することが可能です。FIFO の構成を図 1 に示します。

加速度データのセンシングとデジタル化が完了した後で、サンプルが出力フィルタから出力されます。このデータは FIFO 内部の、データ・レジスタに最も近接したレベルに格納されます。データ・レジスタからの読出し動作時には、FIFO[0]のデータが取得され、FIFO から取り出されるので、残りのデータ・サンプル・セットはデータ・レジスタに 1 レベル近い場所にシフトされます。つまり、FIFO[0]の読出し終了後に FIFO[1]のサンプルが FIFO[0]にシフトし、FIFO[2]のサンプルが FIFO[1]にシフトするという動作です。読出し動作の後で FIFO[0]にシフト可能な新しいデータ・サンプル・セットが存在しない場合には、出力フィルタで新しいデータが有効になるまで古いデータが保持され、新しいデータが出力フィルタから出力された時点で FIFO[0]のサンプルを置き換えます。FIFO が満杯（つまり 32 レベルすべてにデータ・サンプル・セットが格納されている場合）時の出力フィルタの動作は、FIFO の動作モードに応じて異なります。

	X-AXIS		Y-AXIS		Z-AXIS	
OUTPUT FILTER	DATAX0	DATAX1	DATAY0	DATAY1	DATAZ0	DATAZ1
FIFO[31]	DATAX0[31]	DATAX1[31]	DATAY0[31]	DATAY1[31]	DATAZ0[31]	DATAZ1[31]
FIFO[30]	DATAX0[30]	DATAX1[30]	DATAY0[30]	DATAY1[30]	DATAZ0[30]	DATAZ1[30]
⋮	⋮	⋮	⋮	⋮	⋮	⋮
FIFO[2]	DATAX0[2]	DATAX1[2]	DATAY0[2]	DATAY1[2]	DATAZ0[2]	DATAZ1[2]
FIFO[1]	DATAX0[1]	DATAX1[1]	DATAY0[1]	DATAY1[1]	DATAZ0[1]	DATAZ1[1]
FIFO[0]	DATAX0[0]	DATAX1[0]	DATAY0[0]	DATAY1[0]	DATAZ0[0]	DATAZ1[0]
DATA REGISTER (0x32 TO 0x37)						

08234-001

図 1. FIFO バッファの構成

データ・レジスタからのデータの読出し

データ・レジスタからの読出し動作はすべて、複数バイト読出しとして実行することを推奨します。複数バイト読出しとは、複数バイトの最終バイトが取得されるまで通信が終了しない読出しです。複数バイト読出しを行うことで、レジスタから読み出したデータの時間的な同時性を確保することができます。単一バイト読出しを実行すると、1 つの読出し動作が終了した後で次の読出し動作が開始されるまでの間に新しいデータが出力フィルタから出力される可能性があるため、サンプル間の時間的同期が確保できない可能性があります。

SPI 3 線または 4 線通信の場合に複数バイト読出しを行うには、マルチバイト・ビットを設定し、次にバイトごとに 8 個のクロック・パルス・セットを送信し続けます。図 2 に示すように、複数バイト・ビットが設定された状態で、8 個のクロック・パルス・セットがそれぞれ送信された後でレジスタ・ポインタは次の値にシフトします。複数バイト・ビットの場所の説明は、ADXL345 データシートを参照してください。

I^2C 通信の場合、複数バイト読出しは読出しの実行に必要な I^2C プロトコルに則り、SPI の場合の複数バイト読出しと同じ方法で実行されます。読出しを開始した後で、9 個のクロック・パルス・セット(I^2C 通信ではアクノレッジ・ビットとして 9 番目のクロック・パルスが必要)の送信を続けると、レジスタ・ポインタが次の値にシフトします。I^2C 通信での読出し動作の実行に関する詳細については、ADXL345 データシートまたは NXP Semiconductors 社から入手可能な *UM10204 I^2C バス仕様書およびユーザ・マニュアル*、Rev. 03—19(2007 年 6 月発行)を参照してください。

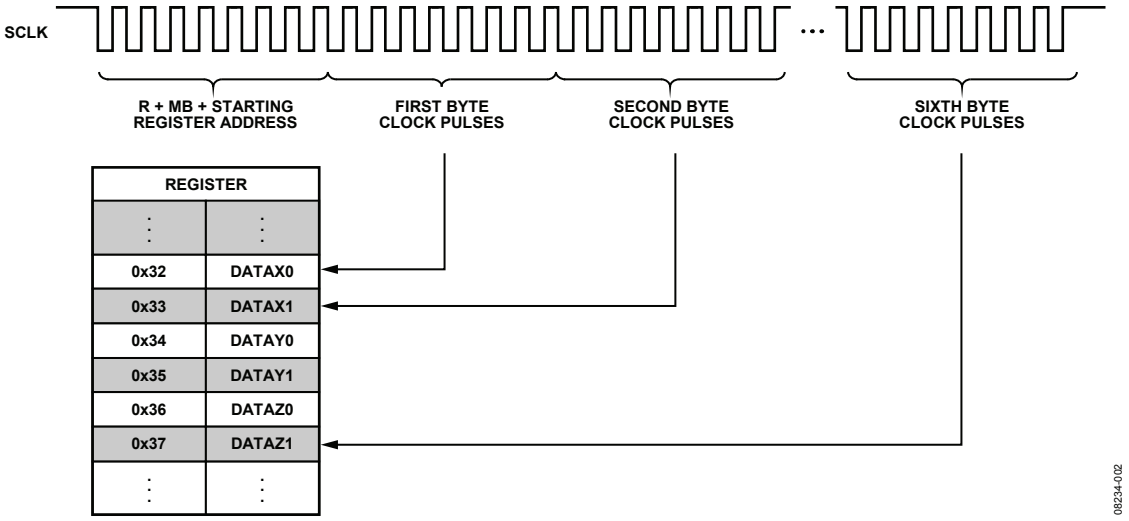

図 2. ADXL345 の複数バイト SPI 読出し時におけるレジスタ・ポインタのシフト

FIFOからのデータの読出し

FIFOの使用中にデータ・レジスタから読出しを行うと、FIFO[0]に格納されたデータ・サンプル・セットを読み出せます。古いFIFO[0]値を取り出して、データ・サンプル・セットをシフトダウンする (これはFIFOのポッピングとも呼ばれます)には、DATAZ1 レジスタの最後のクロック・パルスを受信するか、または通信を終了する必要があります。SPI通信の実行中に通信を終了させるには、$\overline{CS}$ピンのアサートを解除します。I^2C通信の場合はマスター・デバイスからの停止コマンドの発行によって通信を終了します。次のサンプルを読み出すには、図 3 に示すように複数バイト読出しを再度実行してください。

データ・レジスタからのデータの読出しと同様に、2 バイト以上のデータを FIFO から読み出す場合には、複数バイト読出しを行う必要があります。FIFO の使用時に単一バイト読出しを実行すると、読出し動作の終了によって FIFO がポッピングするため、1 つの読出し動作が終了した後で次の読出し動作を開始するまでの間にデータ・サンプル・セットがシフトし、異なるデータ・サンプル・セットからの読出し値が混在してしまいます。たとえば、x 軸の LSB の読出しに単一バイト読出しを使用し、さらに x 軸の MSB の読出しにも単一バイト読出しを使用すると、この読出し動作が終了した後で次の読出し動作を開始するまでの間に FIFO がポッピングし、同じデータ・サンプル・セットからではなく隣接するデータ・サンプル・セットからバイトを返す結果が生じます(図 4 を参照)。

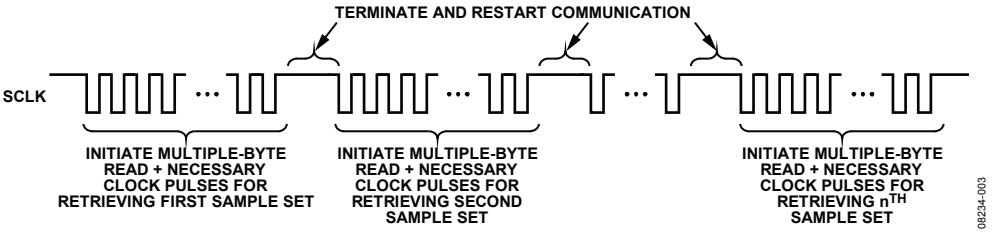

図 3. FIFO の読出し動作が終了した後で次の読出し動作を開始するまでの間に通信の終了と再開を行い、FIFO をポッピングする

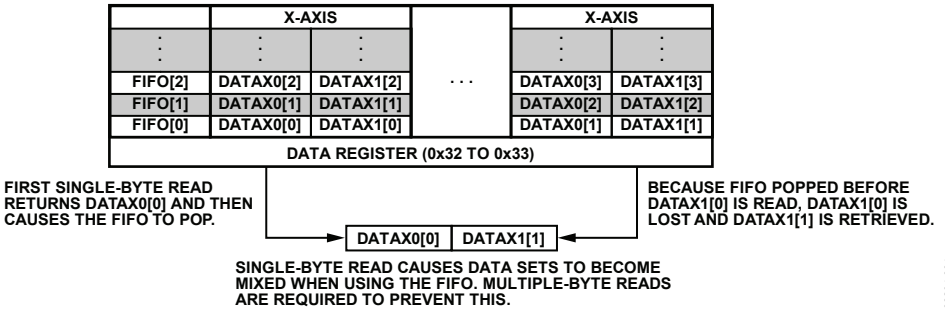

図 4. FIFO 使用時の単一バイト読出しによるサンプルの混在

通信速度、データレート、終了時間

データをFIFOから読み出すことが可能な速度は通信速度に依存するため、データの読出し速度を上回る速度でFIFOを満杯にしないように、適切なデータレートを選択するよう注意してください。I^2Cプロトコルでは、複数バイト読出しを開始するだけのために、Start Condition送信、スレーブ・アドレスおよびレジスタアドレス指定、Start Condition再送信、その後に続くスレーブ・アドレスおよび読出しの動作が要求されます。I^2C通信でのホスト・コントローラの効率もデータの読み飛ばしなく使用できる出力データレートに影響を及ぼすので、出力データレートを決める際にはこの点も考慮に入れてください。一般にI^2C通信を使用する場合は加速度センサーの出力データレートを 800Hz 以下に設定することを推奨します。

SPIでは読出しの開始に関する条件がI^2Cとは異なるため、I^2Cよりも低いクロックレートでも同じ出力データレートでの読出しが可能です。システム設計者は、バス速度に対応する適切なデータレートが選択されていることを検証する必要があります。

連続的な各サンプル読出し動作の間に必要な時間(図 3 に示す終了および再開通信時間)は約 5 µsです。これは、各読出しの間でFIFOがポッピングする必要があるためです。SPI通信の場合は、DATAZ1 の読出しまたは$\overline{CS}$ピンのアサート解除が終了した後で次のサンプル・セットの読出しが開始されるまでの間を約 5 µs以上空ける必要があります。I^2C通信の場合、読出しの開始に必要な高速モードでの最小時間(70 µs)は、この 5 µsの条件を十分に満足する時間です。

5 MHzのSPI通信の場合、読出しの開始に必要な時間はわずかに約 1.6 µsであるため、各読出し動作の間で$\overline{CS}$ピンのアサートを解除している時間を 3.4us以上にすることが要求されます。1.6 MHzのSPI通信の場合は、読出しの開始に少なくとも 5 µsかかるので、SPI速度が 1.6 MHzよりも低い場合には、$\overline{CS}$ピンのアサートの解除に要する時間はSPI通信の仕様として規定される$t_{CS,DIS}$のみになります。

FIFOステータスのモニタリング

FIFOのステータスのモニタ用として、FIFO_STATUS レジスタを使用できます。FIFO_TRIG ビットの説明は「トリガ・モード」に記載しています。Entries ビットは、現在 FIFO に保持されているサンプルの数に対応します。Entries ビットは 6 ビット長ですが、最大許容値は 0x20 または 32(10 進数)です。FIFO からのデータ読出し終了直後に FIFO_STATUS レジスタの読出しを行うと、エントリの値はデータ・サンプル・セットのシフト終了後まで更新されないために不正確となります。データ・レジスタから読出しを開始し、FIFO_STATUS レジスタの読出し後に動作を終了する複数バイト読出しを実行する場合には、DATAZ1 の読出しが終了した後で FIFO_STATUS の読出しが開始されるまでの間に、FIFO のポッピングを可能にするために 5 µs の最小時間が要求されます。この動作は、5 µs の最小時間を満足するまでクロックを引き延ばすか、または停止させた後で通信を再開することで実現できます。

表 1. FIFO_STATUS レジスタ(読出し専用)

D7	D6	D5	D4	D3	D2	D1	D0
FIFO_TRIG	X[1]	Entries					

[1] X = don't care.

FIFOの設定

FIFOの設定は、FIFO_CTLレジスタを通して行います。FIFO_CTLレジスタのFIFO_MODEビットを使用して、FIFOの動作モードを決定および設定します。また、Samplesビットを用いてをウォーターマーク割込みやトリガ割込みの設定を行います。Samplesビットの動作は選択したFIFOの動作モードによって異なります。

表2. FIFO_CTLレジスタ(読出し/書込み)

D7	D6	D5	D4	D3	D2	D1	D0
FIFO_MODE		Trigger	Samples				

誤った割込みの発生、または偶発的なFIFOへのデータ投入を防止するために、対応する割込みをイネーブルにする前、およびデバイスを測定モードに設定する前に、FIFOおよびトリガ・モードでのトリガ割込み、あるいはウォーターマークの設定を行ってください。原則として、デジタル加速度センサーを以下の順に設定します。

1. データレート、計測範囲、データ・フォーマット、オフセット調整などのデータ・パラメータの設定
2. 割込みの設定(イネーブルにしません) : スレッショルドとタイミング値を設定し、使用する割込みをピンを選択します。
3. FIFOの設定(使用する場合) : モード、トリガ・モードを使用している場合はトリガ割込み、Samplesビットを設定します。
4. 割込みのイネーブル : INT_ENABLEレジスタ
5. デバイスの測定モードへの設定 : POWER_CTLレジスタ

この順番に従うと、FIFOが空のとき、およびSamplesビットが0のデフォルト値に設定されるときの誤ったウォーターマーク割込みに加えて、割込みがデフォルト値でイネーブルになっていることに起因する誤った割込みの発生も防止されます。

FIFOモード

FIFOには、バイパス・モード、FIFOモード、ストリーム・モード、トリガ・モードの4つの動作モードがあります。この各モードは、FIFO_CTLレジスタのFIFO_MODEビットを使用して設定します。どの値がどのモードに対応するかは、ADXL345データシートを参照してください。

バイパス・モード

バイパス・モードのときには、FIFOは実質的にディスエーブルになります。出力フィルタから新しいデータが出力されるたびに、データ・レジスタまたはストアされた値が即時に新しいデータ・サンプル・セットに置き換えられます。新しいデータが出力フィルタから出力される前にデータ・レジスタの読出しを再度行うと、古いデータをもう一度読み出せます。新しいデータが有効になるまで古いデータが保持され、新しいデータが出力フィルタから出力された時点でデータ・レジスタ内の古いデータが新しいデータに置き換えられます。新しいデータが有効になる瞬間に読出しの通信を行っている場合には、ダブルバッファ出力により、読出しを最初に終えた後で新しいデータを出力フィルタに送り込み、読出しによるデータ損失を防止します。

このモードでは、新しいデータの読出し準備が完了していることをマスター・デバイスに通知するために、通常はDATA_READY割込みが使用されます。DATA_READY割込みの設定は、選択した割込みピンに対してINT_MAPレジスタでDATA_READY割込みの出力ピンを設定し、次にINT_ENABLEレジスタでDATA_READYビットをイネーブルにします。DATA_READYをマッピングした割込みは、新しいデータが有効になるときに発生し、データ・レジスタからのデータ読出しによってクリアされます。DATA_READY割込みの極性はアクティブ・ハイレベルにデフォルト設定されています。DATA_FORMATレジスタでINT_INVERTビットを設定することにより、DATA_READYと他のすべての割込みをアクティブ・ローレベルに設定できます。

表3. INT_MAP/INT_ENABLEレジスタ(読出し/書込み)

D7 DATA_READY	D6 SINGLE_TAP	D5 DOUBLE_TAP	D4 Activity
D3 Inactivity	D2 FREE_FALL	D1 Watermark	D0 Overrun

FIFOモード

FIFO モードのときに、FIFO は満杯になるまで新しいデータを収集した後で、FIFO に余裕スペースが利用できる状態になるまで新しいデータの収集を停止します。データの読出しによって余裕スペースが利用できる状態になると、出力フィルタから出力された最新のデータが再び FIFO に格納され始めます。FIFO が完全に満杯の状態になり、データが十分迅速に読み出されない場合には、満杯時に FIFO にストアされたデータと、余裕スペースが利用可能な状態になった後に FIFO に格納された新しいデータとの間で、データの時間的な断続が発生します。これを図 5 に示します。FIFO モードの通常の動作方法は、DATA_READY をディスエーブルにし、代わりに Water Mark 割込みを用いる手法です。FIFO モードでは、FIFO_STATUS レジスタの Entries の値に相当する FIFO に格納されたデータの数が、FIFO_CTL レジスタの Samples に書き込まれた値と等しくなると、Water Mark 割込みが発生します。DATA_READY 割込みではなく、Water Mark 割込みを使用すると、所望のレベルまで FIFO にデータを格納し、その後の連続的な複数バイト読出しで FIFO を空の状態にすることが可能であるため、加速度センサーからの割込み発生頻度を低減することができます。Water Mark 割込みを設定するには、ゼロ以外の値を Samples に書き込みます。使用する値は任意に選択可能ですが、割込みの発生時にシステムが割込みをどの程度迅速に処理できるかに応じて異なります。処理時間が十分短く、新しいデータが FIFO に格納される前に FIFO の読出しを終了できる場合には、32 に近い値を選択できます。割込みが発生した後でこれを処理するまでに遅延がある場合には、FIFO がデータを放棄する動作を防止するために、Samples を低い値に選択してください。Samples を設定した後で、INT_MAP レジスタで Water Mark 割込みを INT1 または INT2 のいずれかにマッピングし、その後に INT_ENABLE レジスタで Watermark ビットを設定して、ウォーターマーク割込みをイネーブルにします。

DATA_READY 割込みと同様に、Water Mark 割込みはこれを発生させる条件が取り除かれるまでその設定状態が維持されます。つまり、Entires の値が Samples の値よりも大きいか、またはこれと等しい限り、Water Mark 割込みは発生した状態に保たれます。Water Mark 割込みをクリアするには、FIFO 内のデータ数が Samples で設定した値よりも小さくなるまで(データ・レジスタを介して)FIFO からデータを読み出します。ただし、Water Mark 割込みが非常に頻繁にトリガされることによって、FIFO を使用する利点が失われる問題の発生を防止するために、Samples で設定した値と同等のデータ・サンプル・セットを読み出すことを推奨します。

FIFO モードで FIFO がどのように動作するかを示す例を図 6 に記載しています。この例では、Samples ビットが 6 の値に設定されているので、サンプルが FIFO[5]に押し込まれたときに Water Mark 割込みが発生します。さらに 2～3 個のサンプルの後で割込みが処理され、6 個のサンプルが FIFO から読み出され、FIFO 内のエントリが 6 よりも少なくなり、Water Mark 割込みがクリアされます。

任意の FIFO 動作モードから FIFO をバイパス・モードに設定すると、FIFO が空になる点に注意してください。したがって、FIFO をバイパス・モードに設定する前に FIFO 内のデータを読み出してください。デバイスを測定モードからスタンバイ・モードに設定すると、FIFO の内容は保持されますが、新しいサンプルが FIFO に投入されなくなります。そのために、FIFO をクリアせずにスタンバイ・モードから測定モードに設定するとデータの時間的な断続が発生する可能性があります。

	FIFO CONTENTS							
OUTPUT FILTER	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.

09234-005

図 5.　FIFO モードにおける FIFO のデータ投入順（オーバーランによりデータ損失が発生した状態)

ストリーム・モード

ストリーム・モードで FIFO を動作させるときには、新しいデータ・サンプル・セットが連続的に収集されます。FIFO が満杯になると、FIFO[0]内の最も古いデータが放棄され、残りのデータがシフトダウンして、新しいデータが FIFO[31]に格納されます。FIFO が読み出され、新しいデータの余裕スペースが利用可能になるまで、最も古いデータの放棄が続けられます。この動作を図 7 に示します。

ストリーム・モードは、加速度データの読出しタイミングをホスト側で決定するアプリケーションを対象としています。これはボタンを押す動作、または特定時に加速度センサーからの情報を要求するアプリケーション等が該当することがあります。この理由により、ストリーム・モードでは割込みを使用しない動作が可能です。しかし、ストリーム・モードでの Water Mark 割込みは、FIFO モードの場合と同じ方法で動作し、FIFO 内のデータの数が Samples で設定されている値と等しくなったときに割込みが発生します。

ストリーム・モードに関連する機能の一つとして、INT_SOURCE レジスタの Overrun ビットがあります。FIFO が満杯になった後で、新しいデータが有効になりデータを放棄しなければならない場合には、Overrun ビットが設定され FIFO が満杯になった事を示します。これはすべての FIFO 動作モードで同じように動作しますが、FIFO モードとトリガ・モードでは通常、データの損失を防止するために FIFO が満杯になる前にデータが読み出されます。図 7 を例にとると、Overrun ビットは DATA[32]が出力フィルタで有効になり、DATA[0]が放棄された時点で設定されます。このビットは FIFO が読み出された時点でクリアされます。これにより、FIFO が再び満杯の状態になるまで、Overrun ビットは設定されません。

表 4. INT_SOURCE レジスタ(読出し専用)

D7 DATA_READY	D6 SINGLE_TAP	D5 DOUBLE_TAP	D4 Activity
D3 Inactivity	D2 FREE_FALL	D1 Watermark	D0 Overrun

トリガ・モード

トリガ・モードのときに、FIFO はストリーム・モードと同様の動作を開始し、満杯になるまでサンプルを収集した後で、新しいサンプルを保持するための余裕スペースを設けるために最も古いサンプルを放棄します。トリガ・イベントの発生後に、Samples ビットに設定された値に相当する数の最新のサンプルが、放棄されずに残った古いサンプルとともに保持されます。次に、FIFO は FIFO モードと同様の動作を開始し、満杯になるまで新しいサンプルのみを収集した後で、FIFO 内で余裕スペースが利用可能になるまで新しいサンプルを放棄します。サンプル値を 15 とする場合のこの動作を図 8 に示します。

	FIFO CONTENTS							
OUTPUT FILTER	. . .	DATA[4]	DATA[5]	DATA[6]	DATA[7]	DATA[8]	DATA[9]	DATA[10]
⋮	⋱	⋮	⋮	⋮	⋮	⋮	⋮	⋮
FIFO[6]	. . .	X	X	DATA[6]	DATA[6]	DATA[6]	X	X
FIFO[5]	. . .	X	DATA[5]	DATA[5]	DATA[5]	DATA[5]	X	X
FIFO[4]	. . .	DATA[4]	DATA[4]	DATA[4]	DATA[4]	DATA[4]	X	DATA[10]
FIFO[3]	. . .	DATA[3]	DATA[3]	DATA[3]	DATA[3]	DATA[3]	DATA[9]	DATA[9]
FIFO[2]	. . .	DATA[2]	DATA[2]	DATA[2]	DATA[2]	DATA[2]	DATA[8]	DATA[8]
FIFO[1]	. . .	DATA[1]	DATA[1]	DATA[1]	DATA[1]	DATA[1]	DATA[7]	DATA[7]
FIFO[0]	. . .	DATA[0]	DATA[0]	DATA[0]	DATA[0]	DATA[0]	DATA[6]	DATA[6]

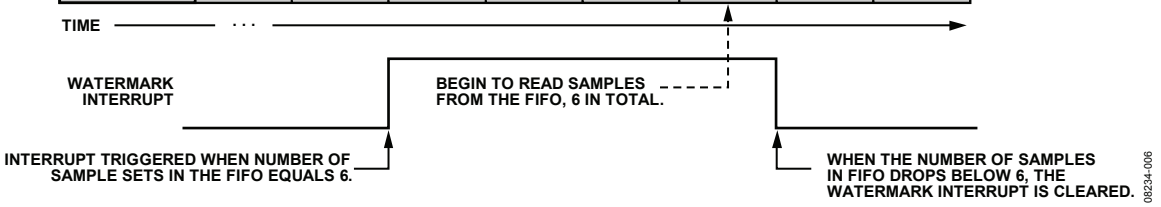

図 6. ウォーターマーク割込みを使用する FIFO モード動作の例、サンプル値 6

	FIFO CONTENTS							
OUTPUT FILTER	DATA[0]	DATA[1]	. . .	DATA[31]	DATA[32]	DATA[33]	DATA[34]	DATA[35]
FIFO[31]	X	X	. . .	DATA[31]	DATA[32]	DATA[33]	DATA[34]	X
FIFO[30]	X	X	. . .	DATA[30]	DATA[31]	DATA[32]	DATA[33]	X
FIFO[29]	X	X	. . .	DATA[29]	DATA[30]	DATA[31]	DATA[32]	DATA[35]
FIFO[28]	X	X	. . .	DATA[28]	DATA[29]	DATA[30]	DATA[31]	DATA[34]
⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋮
FIFO[2]	X	X	. . .	DATA[2]	DATA[3]	DATA[4]	DATA[5]	DATA[8]
FIFO[1]	X	DATA[1]	. . .	DATA[1]	DATA[2]	DATA[3]	DATA[4]	DATA[7]
FIFO[0]	DATA[0]	DATA[0]	. . .	DATA[0]	DATA[1]	DATA[2]	DATA[3]	DATA[6]

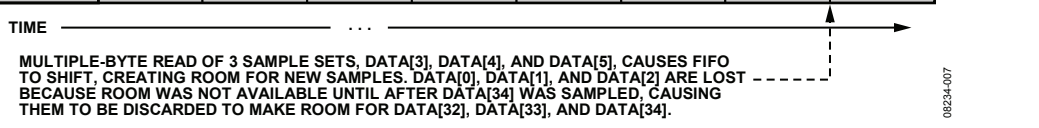

図 7. 最も古いサンプルを放棄するストリーム・モードでの FIFO 動作

	FIFO CONTENTS							
OUTPUT FILTER	DATA[0]	DATA[1]	. . .	DATA[31]	DATA[32]	DATA[33]	DATA[34]	DATA[35]
FIFO[31]	X	X	. . .	DATA[31]	DATA[32]	DATA[33]	X	X
FIFO[30]	X	X	. . .	DATA[30]	DATA[31]	DATA[32]	X	X
FIFO[29]	X	X	. . .	DATA[29]	DATA[30]	DATA[31]	X	X
FIFO[28]	X	X	. . .	DATA[28]	DATA[29]	DATA[30]	X	X
⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋮
FIFO[16]	X	X	. . .	DATA[16]	DATA[17]	DATA[18]	X	DATA[35]
FIFO[15]	X	X	. . .	DATA[15]	DATA[16]	DATA[17]	DATA[34]	DATA[34]
FIFO[14]	X	X	. . .	DATA[14]	DATA[15]	DATA[16]	DATA[33]	DATA[33]
⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋮
FIFO[2]	X	X	. . .	DATA[2]	DATA[3]	DATA[4]	DATA[21]	DATA[21]
FIFO[1]	X	DATA[1]	. . .	DATA[1]	DATA[2]	DATA[3]	DATA[20]	DATA[20]
FIFO[0]	DATA[0]	DATA[0]	. . .	DATA[0]	DATA[1]	DATA[2]	DATA[19]	DATA[19]

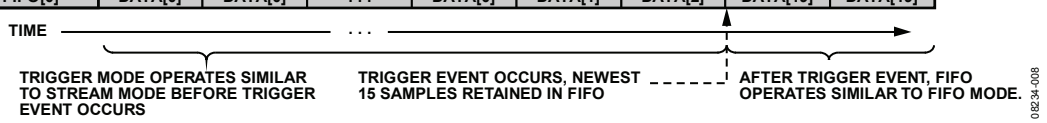

図 8. サンプル値が 15 の場合のトリガ・モードでの FIFO 動作

トリガ・モードに対応するトリガ・イベントは、加速度センサーが持つ任意の割込みです。複数の割込みを選択した場合には、最初に発生した割込みがトリガ・イベントとして認識されます。設定は以下の手順で行います。まずトリガとして使用したい割込みを INT_SORUCE レジスタと INT_MAP レジスタで設定します(ADXL345 データシートを参照)。次に FIFO_CTL レジスタの Trigger ビットでトリガを掛ける割込みピンを選択します。0 の値で ADXL345 の INT1 が選択され、1 の値で INT2 が選択されます。Trigger ビットで選択された割込みピンに設定されている割込みが発生すると、FIFO は Samples の値によって決定されたデータ数を保持した後で、FIFO が満杯になるまでデータの収集を続け、FIFO が満杯になった時点で FIFO モードの動作に遷移します。DATA_READY、Water Mark、Overrun の各割込みをトリガ・モードのトリガ・イベントとして使用しないことを推奨します。

トリガ割込みの発生後は、FIFO が空になるまで読出しを実行してください。FIFO が FIFO モードの動作を継続する場合には、ホスト・プロセッサが加速度センサーからデータを読み出す回数を最小限に抑えるために Samples ビットの値を調整し、ウォーターマーク割込みを「FIFO モード」で説明するように設定する必要があります。これとは別の方法として、ウォーターマーク割込みをトリガ・イベントに使用していない割込みピンにマッピングして、トリガ・イベントが発生した後で初めて FIFO からデータを読み出すように設定することも可能です。その後で、FIFO を FIFO モードで使用する場合には Samples ビットの値を調整し、ウォーターマーク割込みに応答するようにホスト・プロセッサを設定できます。

トリガ・イベントの後に FIFO 内のデータを移動する必要があるため、トリガ・イベントが発生した後で FIFO の読出しを開始するまでに少なくとも 5 µs の時間が必要です。5 µs が経過した後で FIFO は最も古いサンプルを放棄し、保持されたサンプルをシフトします。

トリガ・イベント発生後に再度トリガ・イベントを認識できるようにするには、まず FIFO 内のデータセットを完全に読み出します。トリガ・イベントで発生したデータが必要なければ、FIFO 内のデータを読み出す必要はありません。次に、FIFO_CTL レジスタでデバイスをバイパス・モードに設定し、FIFO 内に残っているすべてのデータを放棄します。FIFO をバイパス・モードに設定した後で再度 FIFO_CTL レジスタの該当ビットを設定して、デバイスを再びトリガ・モードに戻します。この時点でデバイスは次のトリガ・イベントに応答するように設定されます。

FIFOのアプリケーション例

省電力化

FIFO は、ホスト・プロセッサが加速度センサーと通信する回数を低減できるので、特定のイベントが発生するまでデバイスを待機状態にする場合や、データロギングを行う場合にシステムの消費電力を抑えることができます。例としては、輸送する荷物に加わる衝撃のモニタリングや、万歩計(出力の更新がリアルタイムではなくて良い場合)などがあります。

アナログ・デバイセズのデジタル加速度センサーがすでに達成している超低消費電力を考慮すると、ホスト・プロセッサと他の周辺デバイスを適切にスリープ・モードに設定することが、最大の消費電力の節約になります。FIFO を利用することにより、加速度データを連続的に収集し、さらにウォーターマーク割込みを用いて、予め設定した値に基づいて FIFO が満杯の状態に近づいたときに限りホスト・プロセッサがウェークアップするようなシステムを構築できます。FIFO がデータを収集しており、プロセッサが必要とされない時間は、プロセッサをスリープ・モードに設定できるので、システムの消費電力が大幅に低減されます。さらに、他の周辺デバイスを使用している場合には、システムがこれらの動作を要求するまでオフにすることも可能です。

デジタル加速度センサーの FIFO の省電力機能の実例として、図 9 にデータ収集と電流消費の関係を示します。ADXL345 は、100 Hz の出力データレートに設定してあり、プロセッサーとしてアナログ・デバイセズのマイクロコントローラと接続しています。FIFO は 30 サンプルが FIFO 内に格納された時点でウォーターマーク割込みが発生するように設定してあり、データの読出しは SPI で行っています。SPI のクロックは 5MHz に設定してありますので、FIFO 内の 30 サンプルの読出しには約 500µs の時間がかかります。

図 9 の例では、ホスト・プロセッサのウェークアップ、30 のデータ・サンプルの読出し、その後にスリープ・モードに戻る動作に約 3.5 ms の時間がかかります。FIFO の利用による平均消費電流は、ホスト・プロセッサを全時間にわたりアクティブの状態にした場合に必要な 11 mA と比較して、わずか 460 µA に低減されます。これは、96%に近い省電力に相当します。

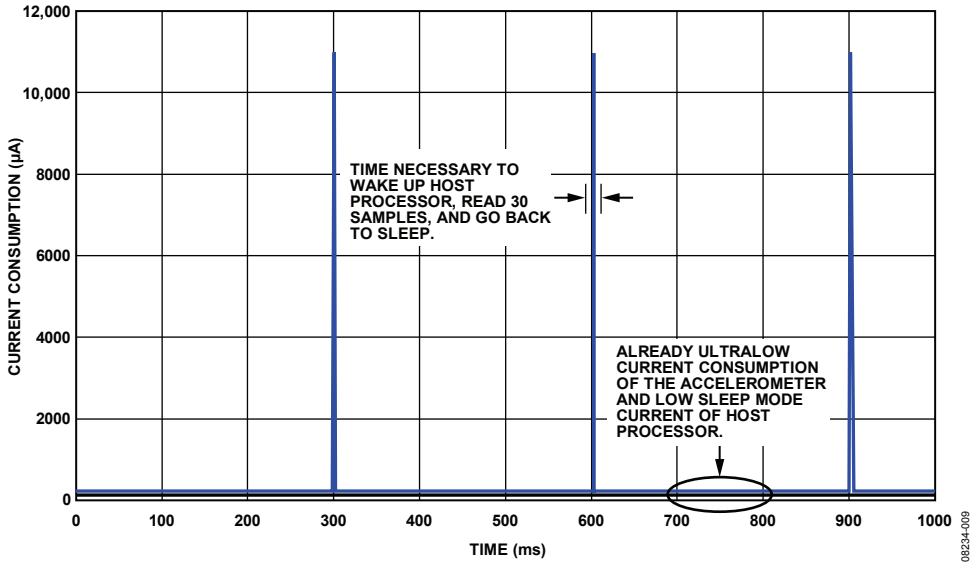

図 9.　FIFO の使用中にホスト・プロセッサをスリープ・モードに設定して達成された省電力

信号処理およびフィルタリング

ADXL345 は約 3.9mg/LSB のスケール・ファクタを持っています。これは約 0.25°の傾斜分解能に相当し、大部分のアプリケーションで十分な数値です。しかしながら、傾斜センシングといった一部のアプリケーションでは、さらに高い分解能とノイズの低減が要求される場合があります。こういった場合には、外部での信号処理が有効です。

低いノイズ、またはさらに高い分解能が望ましい場合には、フィルタリングまたはオーバーサンプリング技術を利用できます。この両方のケースでは、複数のデータを処理する必要があります。これらのアプリケーションでは、必要なデータが溜まるまで FIFO にデータを収集しておくことが可能であるため、処理に必要な量のデータが蓄えられるまでの間ホスト・プロセッサがデータを連続的に読み出して保存する必要がありません。必要なデータがすべて FIFO 内に収集された後で、サンプルのすべてを FIFO から読み込むことが可能であるため、ホスト・プロセッサの負担が軽減されます。

単純な平均化フィルタの場合には、平均化されたデータ数の平方根にほぼ等しいノイズの低減が期待されます。たとえば、100 個のデータを読み込んだ結果のノイズの実効値が 1 LSB(rms)であった場合に、簡単な構成の 4 サンプル平均化フィルタを使用すると、そのノイズが約 0.5 LSB rms に低減されます。例えば、100 Hz の出力データレートで ADXL345 を使用して 100 個のデータ・ポイントを取り込んだデータと、FIFO を使用して 4 つごとにデータを読み込み平均化したデータを比較します。図 10 に示すようにホスト・プロセッサは 4 サンプルごとに一度だけデバイスからデータを読み出すだけで済みプロセッサのデータ読出しの周期は 25 Hz に低減されます。またノイズの低減が確認できます。

AN08234-0-6/09(0)-J

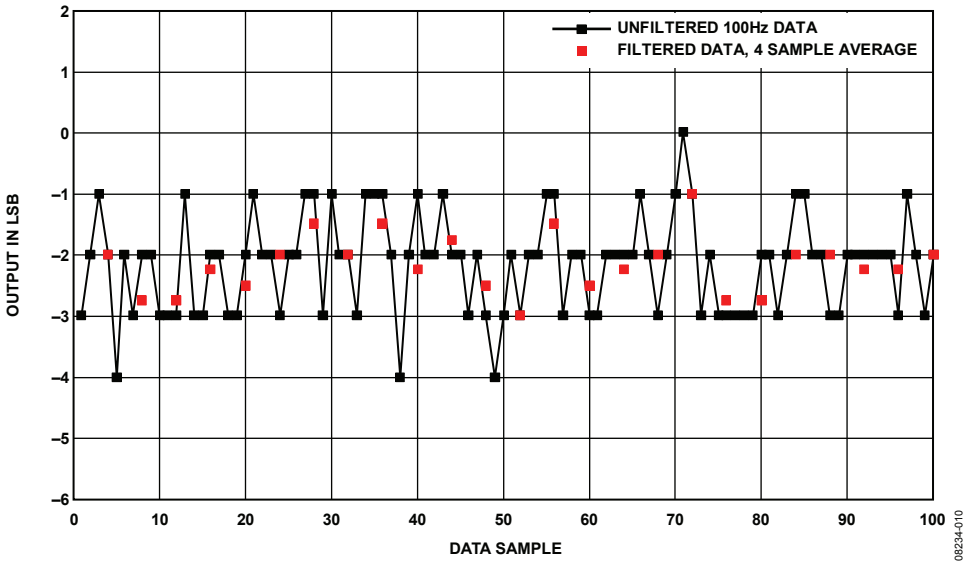

図 10. FIFO を利用した 4 サンプル平均化フィルタによるノイズの低減