

高い精度を実現する連続時間型の $\Sigma\Delta$ ADC【Part 5】 非同期サンプル・レート変換による デジタル・データ・インターフェース の簡素化

著者：Abhilasha Kawle、アナログ設計マネージャ
Naiqian Ren、アプリケーション・エンジニア
Mayur Anvekar、デジタル設計マネージャ

このシリーズでは、連続時間型シグマ・デルタ（CTSD：Continuous-time Sigma-delta）方式のA/Dコンバータ（ADC）について解説しています。CTSD ADCの構成要素である変調器のアーキテクチャは、1つの大きなメリットを提供します。それは、アナログ入力側のシグナル・チェーンの設計を大幅に簡素化できるというものです。今回（Part 5）は、ADCの出力データを、アプリケーションに関連する処理を担うデジタル・ホスト（ADCと併用する外付けのデジタルIC）に伝送するためのシンプルで先進的なインターフェース技術について説明します。ADCのサンプル・レートについては、アプリケーションごとに異なる要件があります。また、どのようなアプリケーションでも、ADCから出力されるデジタル・データの伝送レートはシグナル・チェーンにおける重要なパラメータとなります。

本稿で紹介するのは、ADCの出力に適用する新たなオンチップのサンプル・レート変換技術です。この技術を利用することにより、ADCの出力データ・レート（ODR：Output Data Rate）を、アプリケーションで求められる任意の値に変換することが可能になります。

ADCの役割は、アナログ入力信号をサンプリングし、デジタル形式のデータを生成／出力することです。ADCがアナログ信号をサンプリングする際の周期／周波数をサンプル・レート（サンプリング・レート）と呼びます。一方、ADCの後段では、アプリケーションに必要な処理が行われます。その処理で求められるデータ・レートは、必ずしもADCのサンプル・レートと等しいとは限りません。言い換えると、ADCのODRについては、アプリケーションごとに固有の値が求められます。ADCのサンプル・レートとODRの違いに対応するためのものが、サンプル・レー

ト変換技術（サンプル・レート・コンバータ）です。これを利用すれば、ADCのもともとのサンプル・レートに対応する出力データを、アプリケーションの仕様に応じたODRのデータに変換することができます。

本稿では、まず様々なアプリケーションにおけるサンプル・レートの要件について説明します。それを通して、広範なODRに対応可能なADCの必要性を明らかにします。続いて、ADCで使われている既知のアーキテクチャを例にとり、従来のサンプル・レート変換技術の概要と欠点について簡単にまとめます。その上で、本稿の主題である新たな非同期サンプル・レート変換（ASRC：Asynchronous Sample Rate Conversion）技術を紹介します。この技術を使えば、任意のADCのアーキテクチャと組み合わせて、任意のODRへの変換を実現することができます。その結果、ADCとデジタル・ホストの間のインターフェースの設計を簡素化することが可能になります。特に、CTSD ADCと非同期サンプル・レート変換技術を組み合わせれば、両者の長所を活かし、ADCのアナログ入力側だけでなく、デジタル出力側でもシグナル・チェーンの設計を簡素化することができます。

サンプル・レートに関する要件

ADCを利用するあらゆるアプリケーションでは、その変換精度が重要な意味を持ちます。変換精度は、サンプル・レートを選択する上でも重要なパラメータとして位置づけられます。ADCから出力されるデジタル・データ（サンプル）の数が多くなるほど、もともとのアナログ入力信号をより正確に表現することができます。しかし、デジタル・データの数が多いということは、大量のデータを処理しなければならないということを意味します。そうすると、ADCとデジタル・ホストを接続するためのデジタル・データ・インターフェース（以下、DDI）の設計が複雑になると共に、消費電力が増加するという課題が生じることになります。



このような背景から、各アプリケーションではサンプル・レートを慎重に決定しなければなりません。その際には、必要な精度、消費電力のバジェット、設計の複雑さ、アルゴリズムによって実施する処理内容などに基づいて検討を行う必要があります。一般的に使われるサンプル・レートは、以下に列挙するように分類することができます。

ナイキスト・サンプル・レート

よく知られているサンプリング定理（ナイキスト・シャノンの標本化定理）¹によれば、アナログ入力信号をデジタル・データで正確に表現するためには、入力帯域幅の少なくとも2倍のサンプル・レートでサンプリングを実施する必要があります。多くのアプリケーションでは、この定理を満たすよう、対象とする入力帯域幅の2倍（または2倍強）にサンプル・レートが設定されています。このようなサンプル・レートをナイキスト・サンプル・レートと呼びます。その代表的な例としては、デジタル・オーディオ機器であるCDプレーヤがよく知られています。CDプレーヤで対象とするオーディオ入力帯域幅は最高20kHzです。これは、人間の可聴周波数の上限値に相当します。これに対応し、サンプル・レートとしては44.1kSPS（44.1kHz）が使われています。

オーバーサンプル・レート

周波数領域の高調波の解析や時間領域の波形の解析などを実施する必要があるアプリケーションでは、必要なサンプリング・レートは入力帯域幅の数倍に達します。そのような場合には、図1（b）のようなオーバーサンプル・レートが利用されます。この例では、衝撃を検出するために過渡的な信号を時間領域で解析しています。このような信号をサンプリングする際、図1（a）のようにナイキスト・サンプル・レートを使用すると、山や谷の全体像を把握することができません。より高い周波数でサンプリングを行い、サンプルの数を増やすことにより、信号の忠実な再現と解析が可能になります。

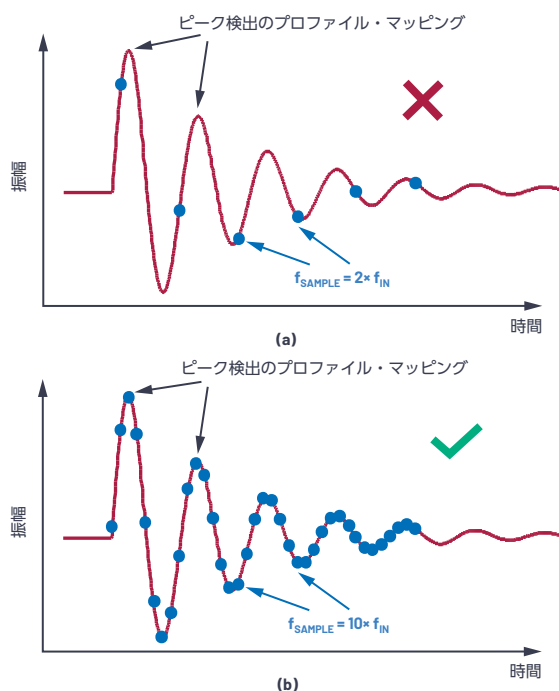


図1. 時間領域における過渡的な信号の解析。
(a) はナイキスト・サンプル・レートを使用した場合、
(b) はオーバーサンプル・レートを使用した場合の例です。

可変サンプル・レート

コヒーレント・サンプリングといった特定のアプリケーションでは、アナログ入力周波数に応じ、サンプル・レートを良好な分解能で調整するという処理が必要になります。そうしたアプリケーションの一例としては、電力ラインの監視が挙げられます。この種のアプリケーションでは、IEC 61000-4-30が定めるクラスAの電力品質メータ規格を満たすために、コヒーレント・サンプリングが使用されます。この規格では、精度に関する要件として、サンプル・レートは入力ラインの周波数ドリフトに追従できるようになっていなければならないと定められています。この種のアプリケーションでは、電力ライン上のクロック・シンセサイザによって、ADC用のサンプリング・クロックを生成します（図2）。

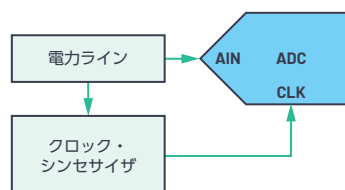


図2. 可変サンプル・レートの使用例。
電力ラインの品質の監視に利用されています。

マルチサンプル・レート

オシロスコープやデータ・アキュイジション・システムなどは、マルチチャンネルのアプリケーションとして実現されます。この種のアプリケーションでは、広範かつ多様なアナログ入力信号の検出／解析を行わなければなりません。そのため、チャンネルごとにサンプル・レートが異なるという状況が発生します。したがって、プラットフォームで使用するADCは、マルチサンプル・レートに対応できるだけの柔軟性を備えている必要があります。

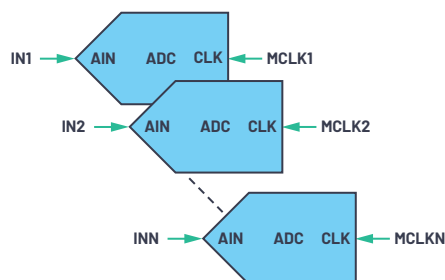


図3. マルチサンプル・レートのアプリケーション

異なる要件への対処

上述したように、ADCのサンプル・レートに関する要件は、アプリケーションごとに異なります。1つのサンプル・レートによってあらゆるケースに対応することはできません。したがって、市場で広く使われるADCを実現するには、プログラムによって多様なサンプル・レートに対応できるようにする必要があります。

図4に、ADCとデジタル・ホストを接続する一般的なDDIの例を示しました。ここで注意すべきなのは、本稿で説明しているDDIには、SPI (Serial Peripheral Interface) やI²Cといったデバイスの構成制御用のインターフェースは含まれていないということです。

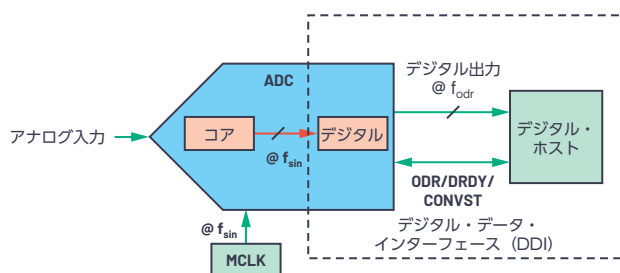


図4. ADC向けの一般的なDDI

図4に示すように、ADCのコア部（以下、ADCコア）では、周波数（レート）が f_{sin} のクロックでアナログ入力をサンプリングします。このサンプリング・クロックは、ほとんどの製品のデータシートではMCLKと表記されています。最終的なデジタル出力のデータ・レートであるODRは f_{odr} です。最も基本的なADCであれば、このODRはサンプル・レートと同じ値になります。また、多くの製品のデータシートを見ると、ODRに対応するクロック・ピンにはODR、DRDY、CONVSTといった名前が付与されています。

ADCコアのサンプル・レート f_{sin} は、ADCのアーキテクチャによって異なります。一方、ODRに相当する f_{odr} は、デジタル・ホストに対するデータ用のインターフェースであるDDIの要件によって異なります。 f_{sin} と f_{odr} については、両者の値が異なったり、両者の間に相関関係も存在しなかったりする可能性があります。そのため、レートが f_{sin} のデータをレートが f_{odr} のデータに変換するサンプル・レート変換の処理が必要になるということです。以下では、ナイキスト・サンプル・レートのADC（以下、ナイキストADC）やオーバーサンプル・レートのADC（以下、オーバーサンプルADC）といった一般的なアーキテクチャで使われている従来のサンプル・レート変換技術について説明します。また、DDIに関する要件についても解説を加えます。

ナイキストADCにおけるサンプル・レート変換

ナイキストADCの場合、ADCコアのサンプリング周波数はアナログ入力帯域幅 f_{in} の2倍に設定されます。ナイキストADCの例としては、最も一般的なタイプの逐次比較型（SAR）ADCが挙げられます。この種の製品では、入力と出力のサンプル・レートは同じ値です。そのため、ODR用のクロックは、ADCコアのサンプル・クロックMCLKとしても使用することができます。一般に、SAR ADCのデータシートでは、デジタル・データの出力に使用するクロック・ピンにはCONVSTまたはDRDYという名前が付与されています。ただ、本稿ではこのピンをODRピンと呼ぶことにします。図5に示すように、ODRピンとMCLKピンを結合すれば、クロック用の配線が1本だけのシンプルなインターフェースを構成できます。このクロックは、外付けのクロック源またはデジタル・ホストから供給／制御されます。つまり、ADCはホステッド・モードで動作することになります。

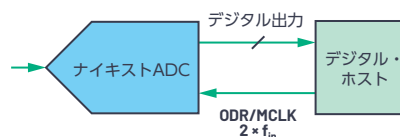


図5. ナイキストADCのシンプルなDDI。
このADCはホステッド・モードで動作します。

アプリケーションの要件とアナログ入力帯域幅に応じて、ODRである f_{odr} をスケールするのは難しいことではありません。その場合、 f_{odr} のスケールに伴い、ADCコアのサンプル・レート f_{sin} もスケールすることになります。また、ADC全体の消費電力もそれに比例してスケールされます。図5に示したようなシンプルなDDIを使えることから、他にも多くのメリットが生じます。1つは、マルチチャンネルのアプリケーションにおいて容易に同期を確立できるというものです。

同期の確立が容易

シングルチャンネルのADCアプリケーションの場合、デジタル・データはADCに供給されるローカル・クロックに対して同期がとれている状態になります。それに対し、マルチチャンネルのADCアプリケーションには、同期に関する2つの課題が伴います。1つは、複数のアナログ入力に対し、確実に同期サンプリングが行えるようにしなければならないというものです。もう1つは、後続のデジタル処理のために、ODRクロックのエッジにデジタル・データを確実に同期させなければならないというものです。同期型のマルチチャンネル・アプリケーションは、ごく一般的に使用されています。代表的な例としては、左右のチャンネルについて同期に関する特定の要件が課せられるオーディオ・アプリケーションが挙げられます。

もう1つの典型的な例としては、電力網における電力ラインの監視アプリケーションが挙げられます。このアプリケーションの場合、各電力ラインにおいて、同期がとられた状態で電圧、電流、入力電力の測定を行わなければなりません。図6に示すように、ナイキストADCを使用する場合、ODRクロックを共有して適切に設計／配線を行うことにより、マルチチャンネルであっても簡単に同期をとることができます。適切な設計／配線とは、ODRクロックが等しい遅延時間で各ADCに伝搬し、可能な限り最良のチャンネル間同期を実現できるようにするということです。

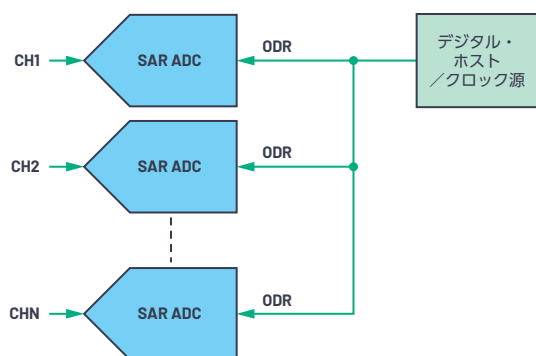


図6. 複数のナイキストADCの同期

ナイキスト・レート制御に関する制約

DDIがシンプルであることは、ナイキストADCの重要な長所です。ここでは、DDIに関連するいくつかの課題について検討しましょう。

ノイズのスケーリング

ナイキストADCでは、アプリケーションのアナログ入力帯域幅に応じて、ODRクロックを簡単にスケーリングすることができます。クロックのスケーリングは消費電力の面では有利ですが、折り返しと呼ばれる現象によってADCの変換結果に含まれるノイズ成分（エイリアス）が増加します。サンプリング定理によれば、ナイキスト周波数よりも高い周波数領域の信号成分は、本来変換の対象としている周波数帯にエイリアスとして現れます。ADCのアナログ入力には、信号源やアナログ入力回路からの不要な情報やノイズが含まれています。それらは、非常に高い周波数まで広がっています。ADCが、それら $f_{\text{sin}}/2$ を超える成分をサンプリングしてしまうと、本来変換の対象としている入力帯域内のノイズが増加するという結果になります。図7に示すように、サンプル・レートを下げるとエイリアスの量が増加し、ADCの出力により多くのノイズが含まれる状態になります。

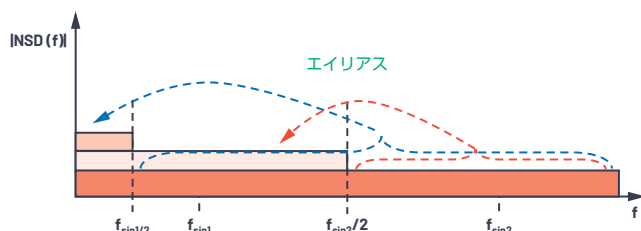


図7. エイリアスとサンプリング周波数の関係

クロックのタイミングに関する制約

SAR ADCの場合、図9 (a) に示すように、サンプリング・クロックには2つのフェーズが必要です。1つはサンプリング・フェーズです。このフェーズでは、ADCの入力部にあるサンプリング用のコンデンサがアナログ入力電圧まで充電されます。もう1つは変換フェーズです。このフェーズでは、サンプリングされた信号がデジタル化されます。ADCの性能を最大限に発揮できるようにするためには、サンプリング回路にはいくらかの最小サンプリング時間を設ける必要があります。サンプリング・クロックを生成するデジタル・ホストやクロック源は、そうしたタイミング関連の制約を満たさなければなりません。

クロック・ジッタ

アプリケーション・ボード上のクロック配線は、クロック源の電源ノイズやボード上の他の信号とのカップリングによる影響を受けます。具体的には、そうしたノイズによって、クロック・エッジの不確実性が増します。この不確実性はジッタと呼ばれています。サンプリング・クロックのジッタは、ADCの性能に影響を与える可能性があります。クロック・ジッタには様々な種類がありますが、最も一般的なのはサイクル間RMSジッタです。これにより、アナログ信号のサンプリング・ポイントが変動し、変換性能が低下します（図8）。このRMSジッタがADCの性能に与える影響については、稿末に示した参考資料2「[Design Note 1013: Understanding the Effect of Clock Jitter on High Speed ADCs](#)（高速ADCに対するクロック・ジッタの影響）」をご覧ください。

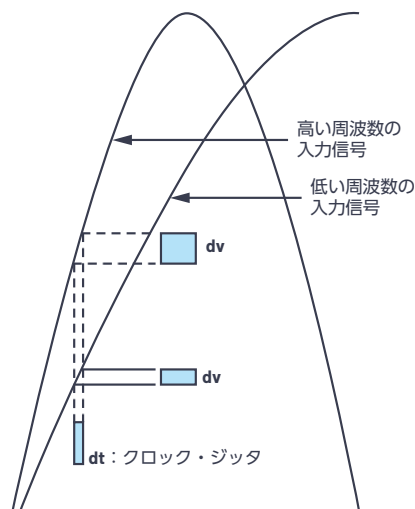


図8. クロック・ジッタの影響。
アナログ入力のサンプリング・ポイントが変動します。

クロック・ジッタによってADCの出力データに付加される誤差は、次式のようにS/N比の低下として定量化することができます。

$$SNR_j = -20 \times \log_{10}(2\pi\sigma_j f_{\text{sin}}) = -20 \times \log_{10}(2\pi\sigma_j \times 2f_{\text{in}}) \quad (1)$$

ここで、 σ_j はRMSジッタです。

上式から、必要なS/N比を得るためには、入力帯域幅を制限しなければならないということがわかります。また、デジタル・ホストやクロック源のノイズが大きい場合には、クロック・ノイズをフィルタリングするための追加の技術が必要になります。

マルチチャンネルのアプリケーションでは、クロック・ジッタはより大きな課題になります。長いクロック配線によって付加されるジッタと同期のバランスをとるためには、優れたクロック・アーキテクチャを用意しなければなりません³。そのようなケースには、ADC向けの低ノイズのクロックを確保するために、適切な絶縁技術とバッファの適用を検討するとよいでしょう。図9 (b) に示すように、絶縁はデジタル・アイソレータを使用することで実現できます。但し、設計の複雑さが増すと共に、消費電力が増加します。

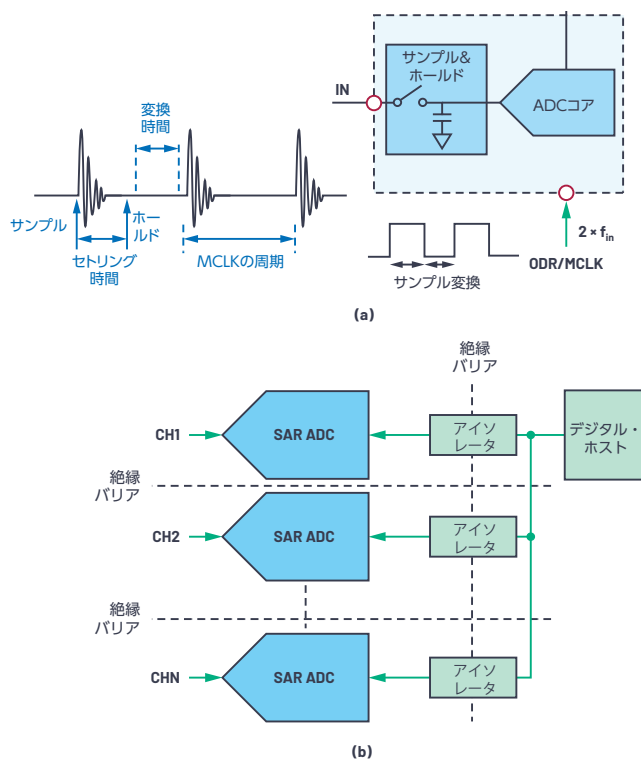


図9. ナイキストADCのDDIの課題。(a) に示すようにクロックのタイミングについての制約が存在します。(b) は、マルチチャンネル・アプリケーションに絶縁を適用する方法を示しています。

オーバーサンプルADCにおけるサンプル・レート変換

続いて、オーバーサンプルADCで使用されるサンプル・レート変換技術について説明します。ADCで連続時間信号をサンプリング／デジタル化する際には、情報の欠落が生じます。出力されるデジタル・データには、必ず量子化ノイズが含まれることになります。ADCにおいてサンプル・レートをナイキスト・レートよりもはるかに高く設定し、より多くのサンプルが得られるようにすると、本来の帯域内の量子化ノイズが低減されて精度が向上します。この技術は、オーバーサンプリングと呼ばれています。高精度のSAR ADCの中には、このオーバーサンプリング技術を採用しているものがあります。

その種の製品は、オーバーサンプリングSAR ADCと呼ばれています。図10 (a) は、オーバーサンプリングSAR ADCにおけるノイズの低減効果について説明したものです。オーバーサンプリング技術を使用する代表的なADCとしては、シグマ・デルタ ($\Sigma\Delta$) ADCが挙げられます⁴。 $\Sigma\Delta$ ADCでは、オーバーサンプリングとノイズ・シェーピングを組み合わせます。ノイズ・シェーピングによって量子化ノイズ Q_e は帯域外に追いやられ、帯域内の性能が向上します。図10 (b) は、その様子を表したものです。サンプリング周波数は、数式で表すと $OSR \times f_{odr}/2$ となります。ここで、OSRはオーバーサンプリング率です。

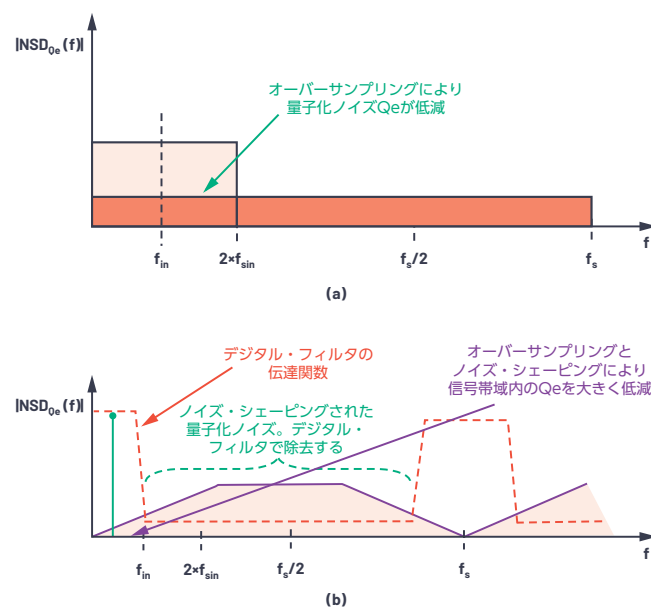


図10. オーバーサンプリングとノイズ・シェーピングの効果。(a) はオーバーサンプリングSAR DACの周波数スペクトル、(b) は $\Sigma\Delta$ ADCの周波数スペクトルを表しています。

オーバーサンプリングADCから出力されたデータをデジタル・ホストに直接伝送したとします。その場合、冗長な情報が多く、ホストを過負荷の状態に陥らせてしまうことになります。場合によっては、ホストはそのような高いデータ・レートでの伝送に伴うタイミング上の厳しい制約に対応できない可能性があります。更に、消費電力が大幅に増加してしまうかもしれません。したがって、ADCからは、要件に応じて最適化された帯域内のデータだけが供給されるようにする必要があります。つまり、アプリケーションの要件に応じ、ODRをナイキスト・レート ($2 \times f_{in}$) またはナイキスト・レートの数倍まで低下させる必要があるということです。このような処理はデシメーションと呼ばれています。デシメーションはサンプル・レート変換技術の一種であり、ADCコアから出力される高いサンプル・レート f_{sin} のデータを、レートが f_{odr} のデータに変換します。

従来から、デシメーションは代表的なサンプル・レート変換技術として使われてきました。最も基本的な手法は、図11に示すように、ADCコアから出力されるデータにフィルタを適用し、 $1/2^N$ にデシメートするというものです。ADCには、サンプリング・クロックMCLKが入力されます。必要となるODR (DRDY)のクロックはMCLKを分周したのですが、これはADCからの出力として供給されます。必要となる分周比は、デシメーション率に応じてNをプログラムするという形で設定できます。プログラミングによって f_{odr} の分解能をより高くするために、アプリケーションの入力帯域幅の要件に応じてMCLKをスケールアップすることも可能です。オーバーサンプリングADCのDDIにおいて、ODRクロックはADCから供給／制御されます。つまり、ADCはホスト・モードで動作します。

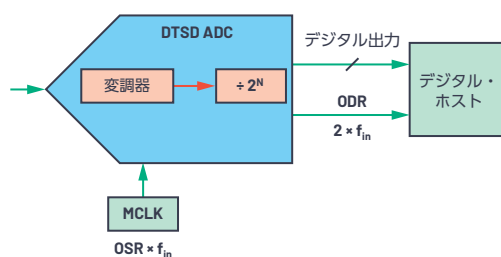


図11. DTSD (離散時間型 $\Sigma\Delta$) ADCのDDI

デシメーションが抱える制約

上述したように、サンプル・レート変換技術としてデシメーションを使用すると、ODRを下げつつADCから高精度のデジタル・データを提供することができます。しかし、この手法には独自の制約があります。以下、それらについて説明します。

非線形なノイズ、消費電力のスケールアップ

可変レートのアプリケーションでは、デシメーション率、MCLK、またはその両方をスケールアップすることになります。デシメーション率だけを高めると、 f_{odr} が低下し、デジタル・フィルタの動きでより多くの量子化ノイズがフィルタリングされてノイズが減少します。このとき、消費電力については、デジタル・フィルタの消費電力だけが直線的に減少します。それに対し、MCLKの周波数を下げると、ADC全体の消費電力が直線的に減少します。但し、その場合には、エイリアスによって帯域内のノイズが増加します。

多くのシステムでは、ADCのMCLKとデシメーション率の両方を調整し、広範なODRを実現します。しかし、この手法では、測定ノイズ性能やシステムの消費電力に望ましくないステップ的な変化が生じる可能性があります。

クロック・ジッタ

オーバーサンプリングADCの場合、サンプル・レート f_{sin} が高いので、式(1)で示したように、ナイキストSAR ADCと比べてクロック・ジッタの影響をはるかに受けやすくなります。そのため、MCLKのクロック源とクロック配線については、アプリケー

ションにおいて許容できるジッタ・ノイズに基づいて設計する必要があります。シングルチャンネルであってもマルチチャンネルであっても、アプリケーション・ボード上には、その全体にわたって多くのスイッチング信号が存在することになります。そうしたノイズの多い信号とMCLKが結合すると、クロック・ジッタが増加してしまうかもしれません。そのため、ADCの性能を最大限に引き出すには、デジタル・アイソレータを使用してMCLKに絶縁を施すという方法を検討する必要があります。しかし、そのような設計を採用すると、実装スペースと消費電力が増大します。プログラミングによって f_{odr} の分解能をより高めるには、MCLKもスケールアップしなければなりません。但し、必要な f_{sin} の値とジッタの要件を満たせるMCLKの発生源を、必ず使用できるとは限りません。

同期

オーバーサンプリングADCが抱えるもう1つの課題は同期です。一般に、 $\Sigma\Delta$ ADCには、同期をとるためのものとしてSYNC_INというピンが設けられています。このピンをトリガすると、アナログ入力の同時サンプリングが開始され、デシメーション・フィルタがリセットされます。デジタル出力データは、デジタル・フィルタのセトリング時間が経過した後に同期がとられた状態になります。図12に示すように、デジタル・フィルタのセトリング中には、デジタル・データの出力は中断されます。また、すべてのADCのMCLKとSYNC_INに入力される信号については、同期がとれていることが前提になります。特にアイソレータやシンセサイザを使用している場合には、サンプル・レートが高い条件下で必要となるすべての同期を確立するのは容易ではありません。

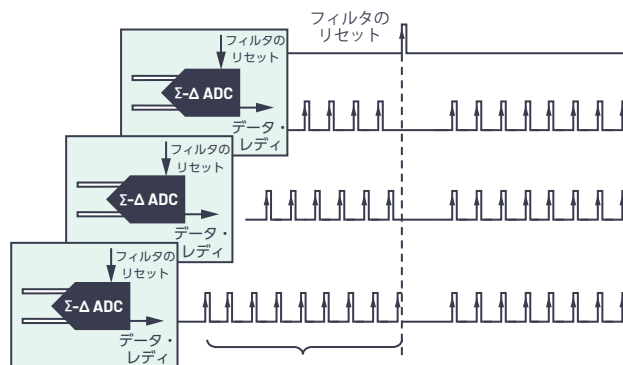


図12. 複数のDTSD ADCにおける同期の確立。
データ出力の中断を伴います。

データ出力の中断と同期の課題の解決に向けたシステム・ソリューションの1つは、すべてのチャンネル向けに、PLLなどのクロック・シンセサイザ回路によって同期をとったMCLKを生成／供給することです。この方法を採用した場合、まずSYNC_INピンがトリガされると、PLLはリファレンス・クロックに対するクロックの同期処理を開始します。PLLのセトリングが行われている最中には、セトリング時間の終了時にADCのサンプリング・エッジとODRクロックのエッジの同期が得られるようにMCLKのレートが調整されます。このソリューションの詳細については、稿末の参考資料5「[ΣΔ ADCの最新アーキテクチャ、分散システムの同期確立時に生じるデータ・フローの中断を回避可能](#)」をご覧ください。

上記のとおり、 $\Sigma \Delta$ ADCやオーバーサンプリングSAR ADCで同期をとるためには、追加の回路やPLL／クロック・シンセサイザが必要になります。SAR ADCを使用する場合と比較すると、設計が複雑になり、消費電力が増加するという点が重要です。

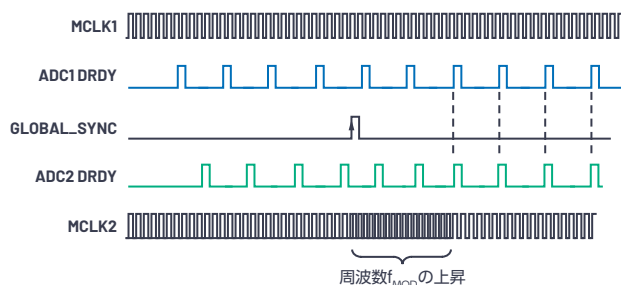


図 13. 複数のDTSD ADCの間で同期をとるためのソリューション。
PLLをベースとした回路でこのような信号／動作を実現します。

同期サンプル・レート変換

上記の課題を解決するために、アナログ・デバイセズは新たな手法を探索してきました。その1つが、同期サンプル・レート変換（SRC：Synchronous Sample Rate Conversion）と呼ばれる技術です⁶。これを使えば、同期に関する課題をある程度軽減することができます。

SRCの長所は、デシメーション率を f_{sin} に対する任意の整数比または分数比に設定できることです。つまり、 f_{odr} のきめ細かい制御が可能になります。アナログ・デバイセズはこの手法について探索し、高精度の離散時間型 $\Sigma \Delta$ （DTSD：Discrete-time $\Sigma \Delta$ ）ADC「AD7770」に適用しました。以下では、SRCの要点に絞って解説を行いますので、詳細については同ADCのデータシートや参考資料などを参照してください。

SRCの重要な特徴は、 f_{odr} を高い分解能でプログラミングできることです。これは、同期をとるのが容易になるということの意味します。例えば、外部入力MCLKを調整することなく、デシメーション率を非常に細かいステップで変化させることができます。図14に示すように、SYNC_INピンをトリガすると、チャンネル間の同期が確立されます。

先述したように、単純なデシメーション手法にはいくつかの制約が存在します。SRCを採用すればMCLKをスケーリングすることなく f_{odr} をより細かく制御できるということは、そうした制約の

ほとんどを解消可能だということを意味します。但し、SRCにも固有の制約と解決すべき課題が存在します。

SRCが抱える制約

すべてのチャンネルで同一のMCLKを使用する場合の同期の課題には、SRCでは対処することができません。

クロック・ジッタ／同期

SRCを使用する場合、MCLKのジッタについては、デシメーションによるサンプル・レート変換を行う場合と同じ制約に直面することになります。 f_{sin} が高い場合、ADCの性能はクロック・ジッタの影響を受けやすくなります。これについては、MCLKに絶縁バリアを適用したり、ノイズを除去するためのフィルタ回路を追加したりすることを検討しなければなりません。マルチチャンネルのアプリケーションではMCLKを複数のADCのチャンネルに配線するので、より問題が大きくなります。マルチチャンネルのアプリケーションで全ADCの同期をとるためには、図16（a）に示すように、MCLKとSYNC_INピンに入力する信号の同期をとらなければなりません。これについては、プリント基板上のクロック信号線の距離や絶縁バリアで生じる遅延とは無関係に、すべてのクロックが各ADCに同時に到達する必要があるがあります。そのためには、配線や絶縁バリアを含めて、クロックに関する設計を注意深く行わなければなりません。信号経路内にアイソレータが配置されていても、すべてのADCで確実に遅延が等しくなるようにする必要があります。

インターフェース・モード

ここまでで説明してきたDDIには、ホスト・モードとホステッド・モードがあります。どちらを使うことになるのかは、ADCコアのアーキテクチャに依存します。例えば、ナイキストADCの場合、デジタル・データ用のクロックは、外部のクロック源やデジタル・ホストから制御／供給されます。そのため、同ADCはプログラミングによってホステッド・モードで動作するように設定しなければなりません。一方、オーバーサンプリングADCは、デジタル・データ用のクロックをデジタル・ホストに供給／制御する役割も担います。つまり、ホスト・モードとして動作するようにプログラムする必要があります。ここまでで説明してきたサンプル・レート変換技術は、いずれも一般的な制約を抱えています。それらを考慮することなくDDIを設計することはできません。

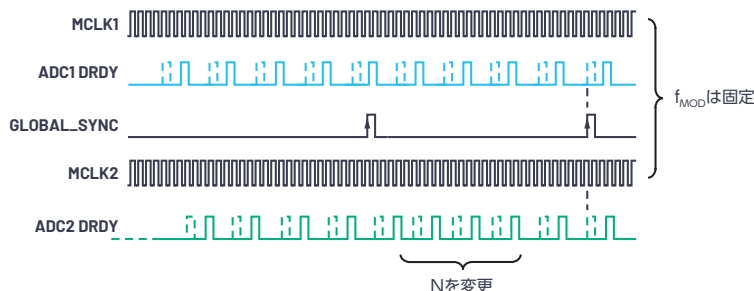


図 14. SRCによるマルチチャンネルの同期

非同期サンプル・レート変換

DDIに関するほとんどの課題に対する解決策は、MCLKに依存する領域とODRクロックに依存する領域を分離することです。その際に重要な役割を果たすのが、非同期サンプル・レート変換(ASRC: Asynchronous Sample Rate Conversion) という新たな技術です。ASRCを適用すれば、MCLKとDDIのクロックを独立させることができます。ODRクロックの選択と制御は、長年にわたりADCコアのアーキテクチャに依存する制限を受けてきました。ASRCは、その障壁を打破する技術です。

ASRCは、ADCコアから出力されるデータを f_{sin} によってデジタル領域でリサンプリングし、それを任意のODRにマッピングするというものです。これについては、任意の非整数によるデシメーションが可能なデジタル・フィルタを思い浮かべていただければ理解しやすいでしょう。性能、サイズ、消費電力の点で最適な実装は、図15に示すように、ASRCで分数デシメーションに対処し、その後に単純なデシメーション・フィルタによって整数デシメーションを実施するというものになります。ASRCでは、ADCコアからのデータをリサンプリングし、それらを $f_{\text{sin}}/N \times f_{\text{odr}}$ でデシメーションします。ASRCの出力のデータ・レートは、 f_{odr} のN倍になります。そのため、デシメーション・フィルタによって $1/N$ のデシメーションを行うということです。

ASRCの実装形態は、図15に示すように2種類存在します。図15 (a) のように、1つ目の実装形態では、ADCからデジタル・ホストに対してODRクロックを供給します。シグナル・チェーンの設計者は、 $f_{\text{sin}}/N \times f_{\text{odr}}$ という係数のプログラムを行うことができます。その際には、ADCの f_{sin} 、必要な f_{odr} 、ADCに実装されたデシメーション・フィルタで使うNの値に基づいて係数を決定することになります。SRCにおけるデシメーション率のプログラミングと似ていますが、デシメーション率として無理数をとることができ、非常に細かい分解能を実現できるという点が大きく異なります。また、SRCの場合と同様に、ODRクロックはMCLKに同期しており、MCLKを分周することによってオンチップで生成/出力されることになります。

ASRCのもう1つの実装形態は、ナイキストADCと同様に、ODRクロックを外部のクロック源やデジタル・ホストから供給するというものになります。その場合、ASRCが備えるクロック・シンセサイザが $f_{\text{sin}}/N \times f_{\text{odr}}$ の比を計算し、ASRCとデシメーション・フィルタに必要なクロックを生成します。ODRクロックは、MCLKに同期している必要はありません。また、ODRの値は任意のサンプル・レートに対し、独立した形で設定することができます。

ASRC技術を採用すれば、実装形態に関わらず、シグナル・チェーンの設計者が f_{odr} の値を細かく設定できるようになります。言い換えれば、 f_{odr} の値がサンプル・レートの整数倍または分数に制限されるという長年の課題が解消されるということです。その結果、ODRクロックのレートとタイミングに関する要件は、純粋にDDIによって決まるようになります。つまり、ADCのサンプル・

レートからは完全に切り離されるということです。上記いずれの実装形態においても、ASRCがもたらす柔軟性により、シグナル・チェーンの設計者にとってDDIの設計は容易になることがわかります。

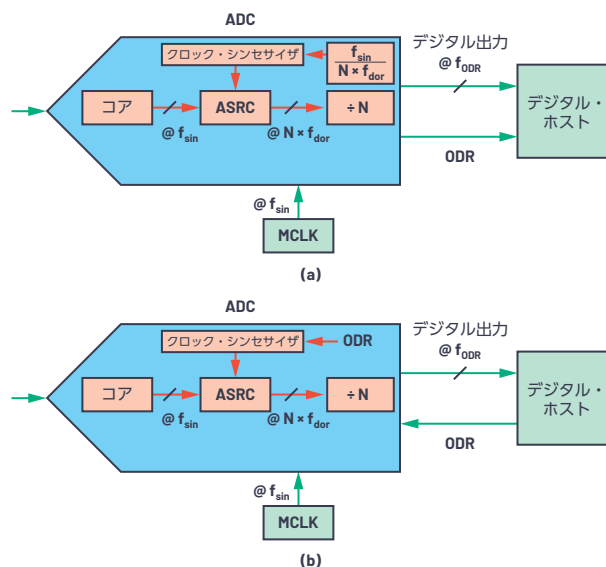


図15. ASRCの実装形態。(a)の実装では、必要な比率を設計者がプログラムします。(b)の実装では、必要な比率はチップ上で計算されます。

ASRCがもたらす価値

ここからは、ASRCがもたらすメリットについてより詳細に解説していきます。

MCLKとODRクロックの分離

ASRCを採用すれば、上記2種類のうちどちらの実装形態でも、 f_{odr} のプログラミング/スケーリングを1Hzの数分の1のレベルで調整できます。このよう細かい分解能を実現できることから、MCLKとODRクロックのレートを独立して選択することが可能になっています。サンプル・レート f_{sin} に対応するMCLKは、ADCの性能とクロック・ジッタの要件に基づいて選択できます。一方、 f_{odr} に対応するODRクロックは、DDIの要件に基づいて設定することが可能です。

クロック・ジッタ

先述したように、ナイキストADCでもオーバーサンプリングADCでも、MCLKとODRには依存関係があります。より細かい分解能で f_{odr} の値を設定するためには、MCLKをスケーリングする必要がありました。しかし、任意の値のMCLKに対し、クロック・ジッタの要件を満たせるクロック源を使用できるとは限りません。そのため、MCLKのジッタによるADCの性能の低下と、 f_{odr} で実現可能な分解能の間には、トレードオフが存在しました。ASRCを採用した場合、ODRとは無関係に f_{sin} の値を選択できます。したがって、MCLKのクロック源としては、できるだけ高いクロック・ジッタ性能が得られるものを選択することが可能です。

インターフェース・モード

ASRCでは、MCLKとODRが分離されているので、インターフェース・モードの選択に自由度が生まれます。ASRCのバック・エンドを搭載したADCは、ADCコアのアーキテクチャに関わらず、ホスト／ホステッド・ペリフェラルとして独立に構成することができます。

同期

先述したマルチチャンネルの同期技術では、MCLKの配線に関して厳しい制約がありました。絶縁バリアとクロックのアーキテクチャは、クロック・ジッタと同期の要件を満たすように設計する必要があります。ASRCを採用した場合、MCLKのクロック源は、図16 (b) に示すように、チャンネルごとに独立させることができます。ホスト・モードの動作では、デシメーション率を独立にプログラムして同期をとることが可能です。一方、ホステッド・モードでは、図16 (b) のとおりODRを共有して同期をとることができます。ODRクロックはデジタル・データのストローブ・クロックにすぎないので、ODRの値は特に高いわけではありません。そのため、MCLKと比べればジッタに関する要件は緩くなります。結果として、絶縁バリアや配線に対する要件も緩和されます。

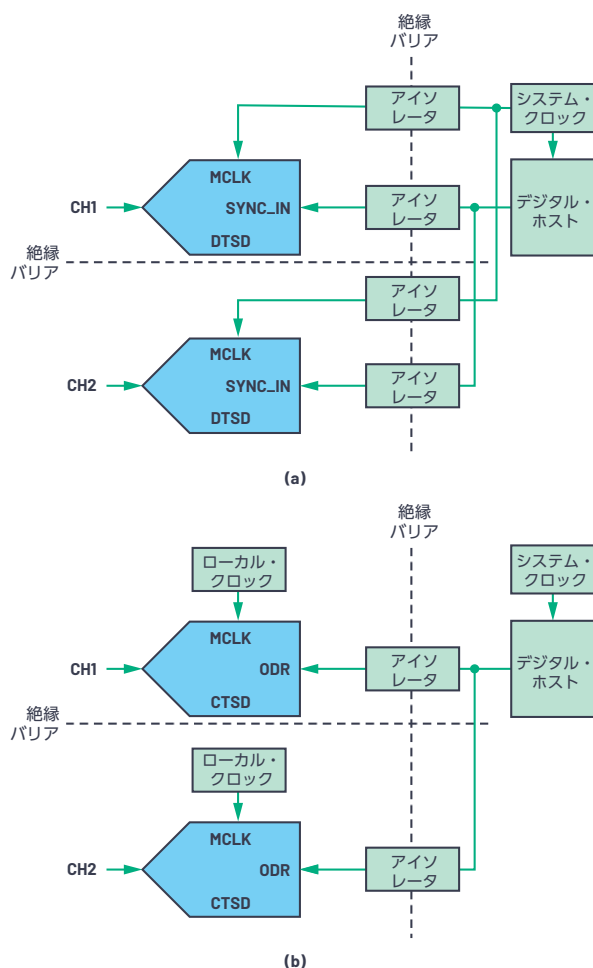


図 16. クロックの分配方法。(a) は、SRC を使用した場合のクロックと SYNC_IN の分配方法を表しています。一方の (b) は、ASRC を採用した場合の例です。クロックの供給と同期の両面で簡素化されていることがわかります。

ASRCとCTSD ADCの組み合わせ

ASRCを採用すれば、デジタル・ホストとのインターフェースを構築するための先進的で簡素化された方法を探求する道が開かれます。また、MCLKとODRクロックを独立させられるという特徴から、ASRCはCTSD ADCにとっての最適な選択肢となります。

DTSD ADCと同様に、CTSD ADCのコアもオーバーサンプリングとノイズ・シェーピングを組み合わせで動作します。また、CTSD ADCのアーキテクチャは、抵抗性の信号入力部／リファレンス入力部、潜在的なエイリアス除去機能といった長所を備えています。そのため、アナログ入力フロント・エンドの設計を大幅に簡素化することができます。本シリーズのPart 2 で説明したように、ADCコアのループは連続時間システムであり、ループの係数はデータシートに規定されている固定のサンプル・レートに合わせて調整されています。

ただ、CTSD ADCには1つ短所があります。それは、DTSD ADCやSAR ADCとは異なり、MCLKがスケラブルではないということです。CTSD ADCをSRCと組み合わせた場合、ODRの値は固定のサンプル・レートに応じて決めなければなりません。それにより、CTSD ADCを使用できる場面が限られてしまうことになります。アプリケーションによっては、固定の f_{sin} に対するODRの比率を無理数に設定したいケースがあります。また、CTSD ADCが最適な性能を発揮するためには、MCLKの周波数精度が高くジッタが小さくなければなりません。具体的には、周波数精度が $\pm 100\text{ppm}$ 、RMSジッタが10ピコ秒といった要件が課せられます。したがって、マルチチャンネル・アプリケーションで付加されるジッタ・ノイズを確実に小さく抑えるために、MCLK向けにはクロック供給用に十分に練られたアーキテクチャが必要になります。MCLKは周波数の高いクロックなので、この課題の解決はより難しくなります。

ASRCを採用すれば、MCLKとODRを分離することができます。そのため、CTSD ADCのアーキテクチャが抱える課題の解消に役立ちます。MCLKのクロック源をローカルでADCの近くに配置することにより、配線が長くなったり、他の信号とカップリングしてジッタ・ノイズが増えたりするのを防ぐことができます。このように、ASRCと組み合わせることで、CTSD ADCのアーキテクチャが備える長所を活かしつつ、固定のMCLKを使用し、そのジッタは低く抑えなければならないという制約に対処することが可能になります。

まとめ

ASRCを採用すれば、ODRを細かく設定できるので、シグナル・チェーンを設計する際の制約を緩和することが可能になります。また、MCLKとODRクロックの依存関係がなくなるので、マルチチャンネルのアプリケーションにおいては、デジタル・アイソレーションに関する設計を簡素化することができます。ADCコアのアーキテクチャに依存することなく、DDIを自由に構成できるので、シグナル・チェーンの更なる簡素化につながります。本稿で述べたように、従来のサンプル・レート変換と比べると、ASRCを採用することによってDDIには様々なメリットがもたらされます。また、ASRCは、任意のADCコアのアーキテクチャと組み合わせることが可能です。特にCTSD ADCと組み合わせた場合には、アナログ入力側とデジタル出力側の両方のシグナル・チェーンの設計が簡素化されます。本稿では、ASRCの必要性ともたらされる価値について説明しました。今後、別の記事において、ASRCの概念をより深く掘り下げ、そのビルディング・ブロックについてより詳しく説明する予定です。それらの情報は、シグナル・チェーンを設計する際、ASRCに関連する性能指標について理解し、その長所をアプリケーションで活かすために役立つはずです。

参考資料

- ¹ Walt Kester 「[MT-002 チュートリアル: 「ナイキストの基準」を、現実のADCシステムの設計に活かす](#)」 Analog Devices、2009年
- ² Derek Redmayne, Eric Trelewicz, Alison Smith 「[Design Note 1013: Understanding the Effect of Clock Jitter on High Speed ADCs \(高速ADCに対するクロック・ジッタの影響\)](#)」 Linear Technology、2006年
- ³ Pawel Czapor 「[クロック・ジッタがΣΔ ADCに及ぼすあらゆる影響を理解する](#)」 Analog Dialogue、Vol. 53、Vol. 2、2019年4月
- ⁴ Michael Clifford 「[Σ-Δ ADCのトポロジに関わる基本原理: パート1](#)」 Analog Devices、2016年1月
- ⁵ Lluís Beltran Gil 「[ΣΔ ADCの最新アーキテクチャ、分散システムの同期確立時に生じるデータ・フローの中断を回避可能](#)」 Analog Dialogue、Vol. 53、No. 3、2019年9月
- ⁶ Anthony O'Shaughnessy, Petre Minciunescu 「[AN-1388 アプリケーション・ノート: AD7779 24ビット同時サンプリング・シグマ・デルタADCを使用した電力品質測定のためのコヒーレント・サンプリング](#)」 Analog Devices、2016年2月

[AD1893 データシート](#)、Analog Devices.



著者について

Abhilasha Kawle (abhilasha.kawle@analog.com) は、アナログ・デバイセズのアナログ設計マネージャです。リニア／高精度技術グループ（インド バンガロール）に所属しています。2007年にインド理科大学院（バンガロール）で電子設計／電子技術に関する修士号を取得しました。



著者について

Naiqian Ren (naiqian.ren@analog.com) は、アナログ・デバイセズのアプリケーション・エンジニアです。入社は2007年で、現在は高精度コンバータ技術グループ（アイルランド リムリック）に所属しています。ダブリン・シティ大学で電気工学の学士号、リムリック大学でVLSIシステムに関する修士号を取得しています。



著者について

Mayur Anvekar (mayur.anvekai@analog.com) は、アナログ・デバイセズのデジタル設計マネージャです。リニア／高精度技術グループ（インド バンガロール）に所属しています。15年にわたり、デジタル回路の設計と検証を担当。組み込みシステムに関する修士号を取得しています。

