

RFトランシーバー用のシンプルな ベースバンド・プロセッサ

著者: Rejeesh Kutty
Analog Devices, Inc.

はじめに

今日、ワイヤレス・システムはあらゆる場所で使われており、ワイヤレス機器やそのサービスの数は増え続けています。フル機能のRFシステムを設計することは、複数の専門分野にまたがる設計課題を包含する作業であり、アナログRFフロント・エンドはその最も重要な部分にあたります。AD9361のような集積化RFトランシーバーを使用することで、このようなRF関連の設計課題は大幅に軽減されます。このトランシーバーはアナログRFシグナル・チェーン用のデジタル・インターフェースを提供し、ベースバンド処理用のASICやFPGAへの統合を容易にします。ベースバンド・プロセッサ(BBP)を使用すれば、エンド・アプリケーションとトランシーバー・デバイスの間のデジタル領域でユーザ・データを処理することができます。また、Simulinkのようなシステム・モデリング・ツールを使用すれば、ベースバンド・プロセッサの設計も容易になります。しかし初心者にとって、この通信システムというパズルのピースを理解し、それを全体の中にはめ込むことは容易なことではないでしょう。本稿は、無線通信システム用の簡素なRFベースバンド・プロセッサの設計と実装に役立つことを意図して書かれています。設計は、AD9361 FPGAリファレンス設計フレームワークを使用して、AD-FMCOMMS2-EBZおよびXilinx® ZC706プラットフォーム上に実装されます。

本稿では、まず始めにこのベースバンド・プロセッサの一般的な設計原理を詳しく説明します。すなわち、BBPの基本的理論について説明します。次いで、アナログ・デバイスのAD9361 FPGAリファレンス設計を使用し、BBPの実際のハードウェア実装について述べます。設計上の主な目標は、設計をできるだけ簡素化し、実験室環境において迅速な無線データ転送をデモンストレーションすることにあります。RFスペクトルを使用し、そのためにRFスペクトルに干渉することは、各種設定に関連し、他に影響を及ぼします。

ベース設計

直接RFシステム以外の代表的なRFシステムを図1に示します。図1に示されているデータ・パスは1つだけですが、逆方向はこのデータ・パスのミラー・イメージです。本稿で説明するベースバンド・プロセッサは、2つのRFシステム間のデータ転送を無線で行いデータを処理することを可能にします。以下では、このベース設計に必要な条件について検討します。

I&Q両方の直交信号でデータを反復

搬送波は互いに独立しており、同期していません。したがって、送信搬送波と受信搬送波の間には、位相と周波数のオフセットがあります。これは、レシーバーでの復調に悪影響を及ぼします。大きな問題となるのは信号の反転で、オフセットが周期的に結合したり分離したりするため、互いに直交する2つの信号の役割が逆転してしまうことがあります。この曖昧な状態を解決する簡単な方法は、2つの直交信号の両方で同じデータを繰り返すことです。

シリアル方式のデータ送受信(ビット単位)

ほとんどの場合、BBPへのRFフロントエンド・インターフェースはDACとADCで、これらはアナログ信号のデジタル・インターフェースです。したがって、データをそのままADC入力に送ることはできません。これはADC出力についても同様です。データはシリアルに送信され、1つのビット・データをDACのフル分解能にマッピングします。同様に、データの受信もシリアルで行われ、ADCのフル分解能からデマッピングされます。これは、高い冗長性を提供します。これらが16ビット・コンバータだとすると、レシーバーは、考え得る65536個のデータ・セットから1か0かを決定します。これだけでも、デコーディングは大幅に単純化されます。

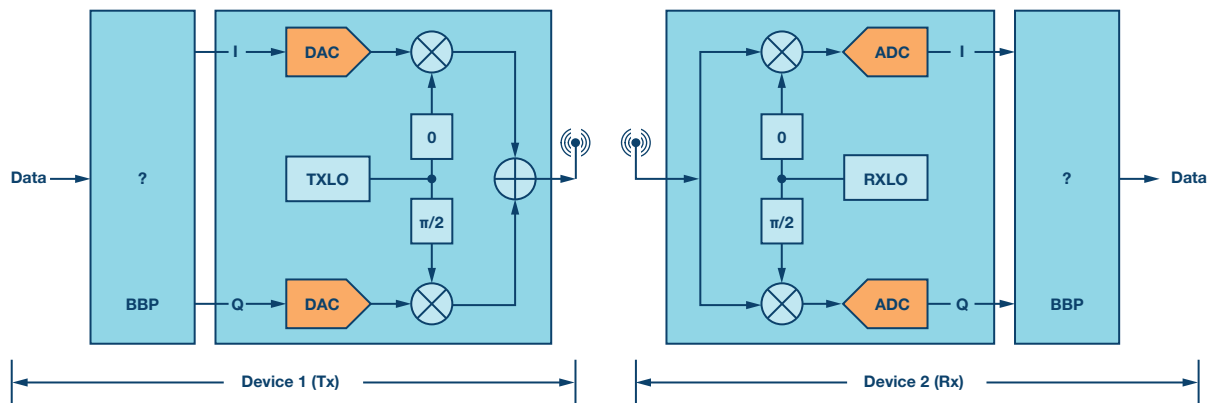


図1. RFシステムのブロック図

互いに直交するI&Q信号

RFフロントエンド・デバイス（AD9361など）は、I/Qトランシーバーです。これらのデバイスは、入力が直交信号である場合に最良の動作をします。通常、これらのデバイスは、2つのデータ・パス間の変動をオフセットするために、両方のデータ・パスに沿って内部的にI/Qマッチングと補正を行います。慣習的に実数 (I) 信号は余弦関数であり、虚数 (Q) 信号は正弦関数です。

BPSKによる変調

一般によく知られているすべての方法、つまり振幅変調、周波数変調、位相変調を展開することが可能です。位相差の検出は比較的容易です。データの転送方式がシリアルなので、BPSK (Binary Phase Shift Keying: 2値位相偏移変調) が自然な選択です。

8サンプルのビット間隔

データにはタイミング情報、つまりビット間隔が必要です。取り得る最大のビット間隔は、サンプリング周期です。レシーバーをできるだけシンプルにするには、信号をデコードして決定を下すための十分な時間が必要です。最も単純なタイミング回復方法は、ゼロ交差とピークの検出です。この場合はピークが一定しないので、ゼロ交差を使用してビット間隔の検出と追跡を行います。また、2つのシステム間には搬送波の差も存在します。場合によっては、ユーザ・データの最初の部分または最後の部分にあるサンプルが不明確なこともあります。正弦波信号の1/2周期ごとに4個のサンプルを取れるようにすれば、ビット間隔は8サンプルに設定されます。したがって、効果的な転送速度はサンプリング周波数の1/8です。

DC成分を含まないデータ

タイミングと相対位相の回復は、信号のゼロ交差に基づいて行われます。したがって、個々の信号にDC成分が含まれないようにする必要があります。また、信号には、ビット間隔ごとに少なくとも1つのゼロ交差が含まれるようにする必要があります。正弦波信号はこれら両方の特性を備えており、上に述べたBPSK変調に非常に適しています。

データのスクランブリング

ユーザ・データは不規則なもので、1または0からなる非常に長いシーケンスとなる可能性があります。データは、レシーバーでタイミングと位相の回復を行って信号をより効果的に追跡できるようにするために、スクランブリングする必要があります。

パケットによるデータ転送

システム同士が互いに同期されていない場合、レシーバーにおける信号には振幅誤差、周波数誤差、位相誤差のあることが予想されます。復調信号は、送信信号の位相のローカル搬送波に対する変化分です。搬送波がしばらく続いた後にデータが現れ、その後再び搬送波が続く場合があります。したがって、設計時にある程度のデータ喪失に備える必要があります。これに対応するために、データの転送はパケットで行います。データ全体を繰り返し転送する代わりに、いくつかのパケットを繰り返し転送することもあります。

CRCによるパケットの妥当性確認

パケットにミスマッチがある場合、レシーバーがそのパケットを廃棄して再送信を要求できるように、パケットには巡回冗長検査 (CRC) 用のデータが付与されます。

プリアンブルごとのタイミング補正と位相補正

パケット・ヘッダーには、受信したデータ・ストリームの中でパケットを識別するためにプリアンブルが付与されます。このプリアンブルは、レシーバーが信号のタイミングと位相情報をリセットして、パケット・データを復調するのに使われます。

組み込み性能指標

レシーバーは、受信パケット数、廃棄パケット数、または修正パケット数といった統計データのカウンタもサポートしています。これらのカウンタは、ビット・エラー・レートや実効データ・レートを含む性能指標の測定とモニタに使われます。

要約すると、データはパケットとしてシリアルで送受信されます。パケットにはプリアンブルとCRCが付与されています。データは、トランシーバー・デバイスの前に、BPSKにより中間直交信号で変調と復調が行われます。中間信号周波数と、それにより決定されるデータのビット・レートは、サンプル・レートの1/8です。以上に概要を示した設計内容のベースバンド・プロセッサ・モジュールを、図2と図3に示します。

トランスミッタは数バイトのデータ (キャラクタ幅) を読み込んで、それをヘッダーまたはプリアンブル付きのパケットに変換します。パケットの末尾にはCRCが追加されます。パケット・データはその後スクランブリングされ、シリアル化されます。更に、トランシーバーとのインターフェースを取る前に、1ビット・データが余弦 (I) 関数と正弦 (Q) 関数を位相変調します。

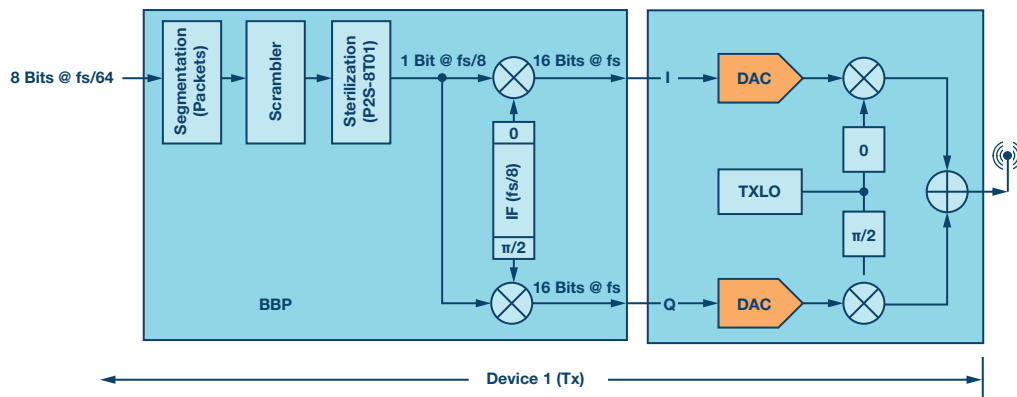


図2. BBP送信機能のブロック図

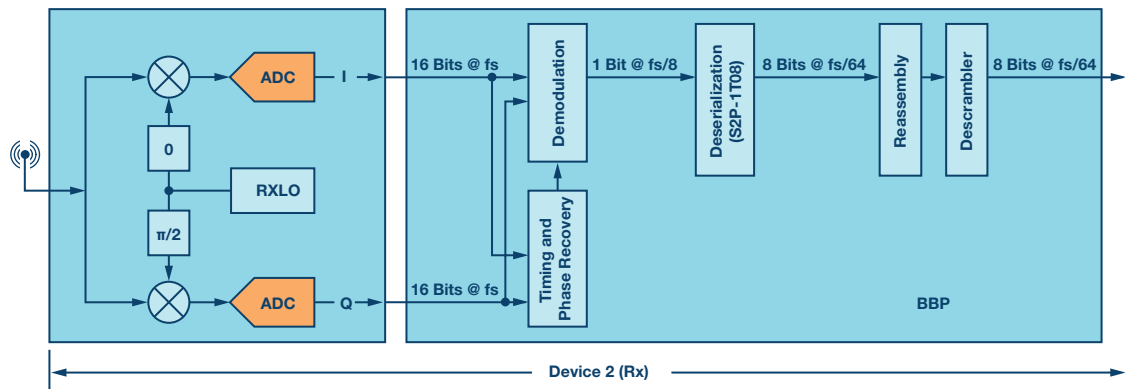


図3. BBP受信機能のブロック図

受信方向では、オフライン・モジュールが、変調信号のタイミング間隔と相対位相の回復と追跡を行います。この情報は、受信ADCサンプルからシリアル・データを回復するために使用されます。その後データをパケットに組み立てて、スクランプリングを解除します。パケットの最後にCRCの比較が行われ、ミスマッチがある場合そのパケットは廃棄されます。CRCがマッチした場合は、エンド・ユーザにデータが渡されます。

実装

BBP設計はハードウェアで実装し、テストします。ハードウェアは2枚の評価用ボードを組み合わせたものです。1枚はZynq FPGAデバイスを搭載したXilinx ZC706評価用ボード、もう1枚はAD9361トランシーバーを搭載したAD-FMCOMMS3-EBZ評価用ボードです。アナログ・デバイセズは、このハードウェアをサポートする機能を完備したリファレンス設計を提供しています。このオープンソース設計は無償で使用可能で、完全なサポートが提供され、主要なツール・バージョンごとに更新されます。ハードウェアの詳細は以下のURLで参照できます。

[ZC706](#)

[AD-FMCOMMS3-EBZ](#)

[Zynq SOC](#)

[AD9361](#)

[アナログ・デバイセズGitHubリポジトリ](#)

[ZC706およびAD-FMCOMMS3-EBZ HDLリファレンス設計](#)

[AXI AD9361 IP](#)

アナログ・デバイセズのリファレンス設計は、Linux®フレームワークをサポートする組み込みシステムで、ARM®プロセッサを中心とする様々な周辺機能で構成されています。AD9361デバイスは`axi_AD9361` IPペリフェラルとのインターフェースを

構成し、RFデバイスとシステム・メモリの間で未加工サンプリング・データの転送を行います。周辺機能とデバイスの初期化および制御は、Linuxのカーネル・ドライバを介して行われます。BBPは、`axi_AD9361`とのインターフェースを構成する別のIPペリフェラルとして実装されます。BBP IPは、これまでの経緯から`axi_xcomm2ip`と呼ばれています。システム間の制御、送信、受信には、Linuxのユーザ・スペース・アプリケーションが使われます。

アナログ・デバイセズのリファレンス設計では、`axi_AD9361` IPが、送信方向パッキング解除モジュール(`util_upack`)と受信方向パッキング・モジュール(`util_cpack`)へのインターフェースを構成しています。送信方向では、パッキング解除モジュールとAD9361コアの間にBBPデータが挿入されます。これがデフォルトのデータ・パスに影響を与えないようにするために、BBPは、パッキングが解除されたデータ・ソースまたはBBPデータ・ソースのどちらかを選択するオプションのデータ・パス・マルチプレクサをサポートしています。BBPを使用すれば、リファレンス設計のデータ・パスをデフォルトにして、BBPデータ・ソースがイネーブルされた場合のみBBPデータ・ソースを選択することができます。受信方向では、BBPは単にAD9361コアへのインターフェースを構成します。リファレンス設計のデータ・パスは影響を受けません。したがって、妨げられることなくシステムを起動してセットアップするフレームワークが可能となります。システムのセットアップ後は、デフォルトのデータ・パスをオーバーライドすることによりBBPがイネーブルされ、データ転送が可能となります。アナログ・デバイセズのリファレンス設計に実装されているBBPのブロック図を図4に示します。

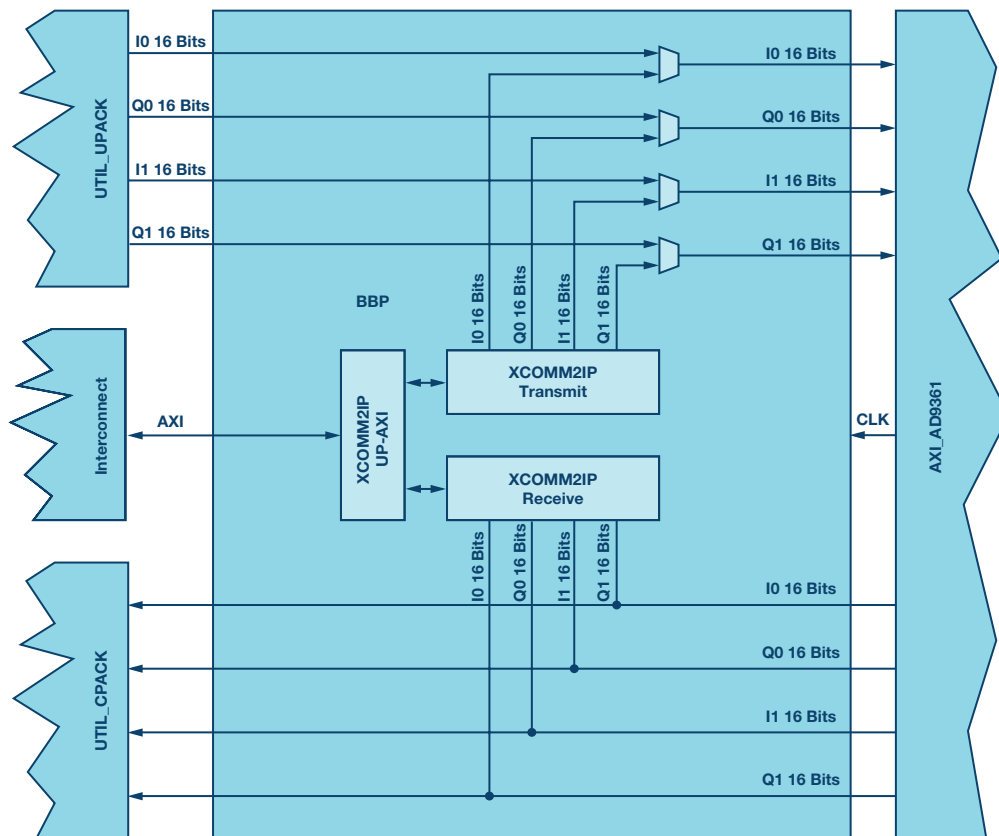


図 4. BBP IPのブロック図

表1. BBPのレジスタ・マップ

アドレス	レジスタ名	名前	説明	タイプ
	フィールド			デフォルト
0x000	XCOMM2IP_VERSION	VERSION	IPバージョン	RO 0x40063
0x008	XCOMM2IP_SCRATCH	SCRATCH	スクラッチ・レジスタ。	RW 0x0
0x800	XCOMM2IP_TX_RESET	TX_RESET	0x1に設定すると、送信がリセットされます。通常動作時は、このビットを0x0に設定する必要があります。	RW 0x1
0x804	XCOMM2IP_TX_ENABLE	TX_ENABLE	0x0に設定した場合はデータ・バスへの影響はなく、UNPACKデータがDACに送信されます。0x1に設定した場合は、BBPデータがDACに送信されます。	RW 0x0
0x808	XCOMM2IP_TX_REQ	TX_REQ	ソフトウェアによって設定された場合は、送信方向でパケットの転送が開始されます。このビットは、転送が完了すると、ハードウェアで自動的にクリアされます。	RW 0x0
0x80C-0x87C	XCOMM2IP_TX_PKT_DATA_3- XCOMM2IP_TX_PKT_DATA_31	TX_PKT_DATA_3- TX_PKT_DATA_31	3～31のパケット・データ・バイト。ハードウェアは最初の3バイトをヘッダー用に使用し、残りのバイトをCRCに使用します。	W W
0xC00	XCOMM2IP_RX_RESET	RX_RESET	0x1に設定すると、受信がリセットされます。通常動作時は、このビットを0x0に設定する必要があります。	RW 0x1
0xC08	XCOMM2IP_RX_REQ	RX_REQ	ハードウェアによって設定された場合、パケットが受信され、ソフトウェアによる読出しが必要であることを示します。パケット・データの読出し後は、ソフトウェアでこのビットをクリアする必要があります。以降のすべてのパケットは、このビットがソフトウェアによってクリアされるまで廃棄されます。	RW 0x0
0xC0C-0xC7C	XCOMM2IP_RX_PKT_DATA_3- XCOMM2IP_RX_PKT_DATA_31	RX_PKT_DATA_3- RX_PKT_DATA_31	3～31のパケット・データ・バイト。ハードウェアは最初の3バイトをヘッダー用に使用し、残りのバイトをCRCに使用します。	W W

本稿で解説している設計、初期化、データ転送は、このハードウェアの組み合わせを使用しています。セットアップに必要なのは、HDMI®モニタ、キーボードとマウス、およびアンテナだけです。システムは互いに完全に非同期ではありますが、同じ設定にする必要があります。データは、それぞれの方向で異なる搬送波により転送されます。デバイス1の送信搬送波周波数とデバイス2の受信搬送波周波数は同じですが、方向が変わると周波数も異なります。ただし、単一のデバイスを使用してループバックを行う場合は、送信搬送波と受信搬送波が同じ周波数でなければなりません。BBPのHDL設計は、アナログ・デバイスズのライブラリ・モジュールを使用しています。

制御(マイクロプロセッサ)インターフェース

プロセッサによるBBPの制御とモニタには、AXI-Liteインターフェースを使用します。このインターフェース・モジュールは、アナログ・デバイスズの共有ライブラリ(hdl/library/common/up_axi.v)にあるup_axiモジュールを使って構成することができます。このモジュールは、AXI-Liteインターフェースを読み出し/書き込みバスのような単純なメモリに変換し、アナログ・デバイスズの他のIPと同様に内部レジスタとメモリが追加されます。レジスタ・マップの詳細は、表1を参照してください。

以下では、up_axiモジュールのポートとそのポート・マッピングについて説明します。

up_rstn : AXIインターフェースのリセット(非同期のアクティブ・ロー)。**s_axi_aresetn**に接続。

up_clk : AXIインターフェース・クロック。**s_axi_aclck**に接続。

up_axi_* : AXIインターフェース信号。等価s_axi_*ポートに接続。

up_wreq、**up_waddr**、**up_wdata**、**up_wack** : 内部書き込みインターフェース。書き込み要求を示すために、アドレスおよびデータと共に**up_wreq**信号がアサートされます。要求に対しては、**up_wack**ポートを介して確認応答を行う必要があります。

簡単なレジスタ書き込みは、以下のように実装されます。

```
always @(negedge up_rstn or posedge up_clk)
begin
  if (up_rstn == 0) begin
    up_wack <= 'd0;
    up_reg0 <= UP_REG0_RESET_VALUE;
  end else begin
    up_wack <= up_wreq_s;
    if ((up_wreq_s == 1'b1) && (up_waddr == UP_REG0_ADDRESS)) begin
      up_reg0 <= up_wdata[UP_REG0_WIDTH-1:0];
    end
  end
end
```

このモジュールは2つのアドレス間の変換を行います。AXIインターフェースはバイト・アドレスを使用しますが、内部バスはDWORDアドレスを使用します。結果として、**up_axi**モジュールはAXIアドレスの下位2ビットを省略して、内部DWORDアドレスを生成します。

up_rreq、**up_raddr**、**up_rdata**、**up_rack** : 内部読み出しインターフェース。読み出し要求を示すために、アドレスと共に**up_rreq**信号がアサートされます。この要求には、読み出しデータと共に、**up_rack**ポートを介して確認応答を行う必要があります。

上記と同じレジスタ読み出し用の実装を以下に示します。

```
always @(negedge up_rstn or posedge up_clk)
begin
  if (up_rstn == 0) begin
    up_rack <= 'd0;
    up_rdata <= 'd0;
  end else begin
    up_rack <= up_rreq_s;
    if ((up_rreq_s == 1'b1) && (up_raddr == UP_REG0_ADDRESS)) begin
      up_rdata <= up_reg0;
    end else begin
      up_rdata <= 32'd0;
    end
  end
end
```

読み出しにも同じアドレス変換が適用されます。読み出しデータは要求がある場合にのみ駆動され、それ以外の場合は0に設定されます。これは、**up_axi**モジュールが、様々なアドレス・ブロックからの個々の読み出しデータをORゲートに渡すためです。したがって、選択されなかったブロックは読み出しデータ0を駆動する必要があります。

上に示すレジスタ・マップの表にリストされているように、BBPには3つのアドレス空間があります。共通レジスタ空間は0x000にマップされ、送信(DAC)は0x800(0x200)に、受信(ADC)は0xC00(0x300)にマップされます。ソフトウェア(Linuxユーザ・スペース・アプリケーション)は、送信パケット・データをバッファに書き込み、受信パケット・データを別のバッファから読み出します。パケット・サイズは、3バイトのプリアンブルと1バイトのCRCを含めて32バイトとなるように選ばれます。

データ・インターフェース

AD9361のインターフェース・コアは、受信および送信方向の2チャンネルに対応する、2組の16ビットI/Qデータで構成されています。このコアは、AD9361デジタル・インターフェースと同じクロックで動作します。これは、2R2Tモードではサンプリング・レートの4倍、1R1Tモードではサンプリング・レートの2倍です。実効データ・レートは有効信号によって制御されます。したがって、2R2Tモードでは4クロックごとに有効が1回アサートされ、1R1Tモードでは2クロックごとに有効が1回アサートされます。BBPは2R2Tモードと1R1Tモードの両方をサポートするように設計されており、1つのチャンネルを使用して送信と受信を行います。内部ロジックは、2R2Tモードでも1R1Tモードでもサンプリング・レートで動作するように作られています。更にBBPはインターフェース・コアを使い、そのクロック周波数でデータを転送します。これは、BBP内でのクロック変換をデモンストレーションするために意図的に行ったものです。多くの場合、BBPロジックは、トランシーバーのインターフェース・レートに関わらずサンプリング・レートで実行することができます。

サンプリング周波数における内部クロックは、XilinxのBUFRプリミティブとBUFGプリミティブを使って生成されます。BUFRは分周器であり、BUFGは高ファンアウトのクロック・バッファです。このためにMMCMを使用することも可能です。内部クロックは以下のように生成されます。

```
parameter XCOMM2IP_1T1R_OR_2T2R_N = 0;
localparam XCOMM2IP_SCLK_DIVIDE =
(XCOMM2IP_1T1R_OR_2T2R_N == 1) ? "2" : "4";

BUFR #(.BUFR_DIVIDE(XCOMM2IP_SCLK_DIVIDE))
i_bufbr (
  .CLR (1'b0),
  .CE (1'b1),
  .I (clk),
  .O (s_clk_s));

BUFG i_bufg (
  .I (s_clk_s),
  .O (s_clk));
```

BUFRとBUFGを使用することでクロックは周波数ロックされますが、位相安定性が犠牲になります。位相の不確実性は、最大でインターフェース・クロック1周期分です。これは、同期信号を使う4段レジスタ・アレイによって簡単に補正できます。ただし、この設計はデュアル・ポートRAMモジュールを使用してデータ転送を行います。これは、共通信号処理要求の使用事例として意図的に行っているものです。デュアル・ポートRAM素子は、アナログ・デバイセズのライブラリに含まれるメモリ・モジュール(ad_mem)を使用して構成できます。

送信インターフェース

送信方向では、プロセッサがバッファにパケット・データを書き込みます(既出のレジスタ・マップの表を参照)。更に、プロセッサはこのパケットを送信するようハードウェアに要求します。BBPIは連続してデバイスへパケットを送信します。BBPIはパケットの冒頭で要求をチェックし、保留要求がない場合はアイドル・パケットを送信します。保留が要求された場合は、パケット・バッファが読み出されて送信されます。

送信ロジックは、自走ビット・カウンタを使いビット幅で実行されます。バッファの読出しアドレスは、ビット・カウンタが0x0のときに更新されます。プロセッサの要求はパケット転送中の任意の時点で行われるので、パケット転送開始時に直ちにキャプチャされてクリアされます。パケット転送開始時に要求が保留されている場合は、その旨の確認応答がプロセッサ・インターフェースに返されます。この要求は、バッファ・データまたはアイドル・データを選択するために使用されます。

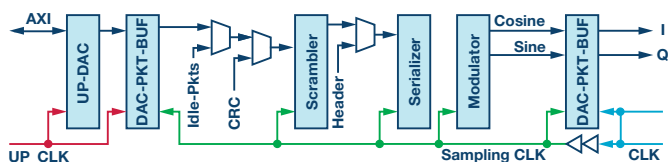


図5. 送信データ・パス

パケット・データの最初の2バイトは、0xffff0に設定されます。3番目のバイトは、アイドル・パケット(0xc5)かデータ・パケット(0xa6)かを示すために使用されます。CRCバイトはパケットの最終バイトとして挿入されます。CRC多項式は、 $x^8 + x^2 + x + 1$ です。ヘッダーを除くすべてのバイトはスクランプリングされます。スクランプリング多項式は、SONET/SDH($x^7 + x^6 + 1$)と同じです。

変調搬送波の生成には、余弦および正弦ルックアップ・テーブルが使われます。ビット間隔は、8サンプルの信号のフル・サイクル(0~ 2π)に等しくなります。ビット・データは信号の反転に使われます。その後データは小さいバッファに書き込まれ、インターフェース・クロックを使い、AD9361インターフェース・コアからの有効信号に基づいて読み出されます。

受信インターフェース

受信方向では、ヘッダー・パターン0xffff0のI/Qデータがモニタされます。これは、1回のパケット転送に1回だけ出現する固有のパターンです。スクランブラ出力がこのパターンを繰り返すようにしてパケット・データを送信することもできますが、これは推奨されないため、ソフトウェアで回避します。12個の連続したビット間隔に対するこの一連の同相データ・シーケンスは、タイミング回復モジュールがレシーバーのタイミングと位相をリセットして追跡するために使われます。したがって、そのタイミング・カウンタをリセットして位相値を0x1に設定します。このシーケンス後の最初の反転は、0x0と見なされます。この後のタイミング回復モジュールは、パケット転送の全期間を通じてその状態を維持します。

データ回復モジュールは、信号を平均して現在の信号の位相を決定します。更にこの位相が、タイミング回復モジュールの追跡する相対位相と比較されます。両者が競合する場合は、過去の信号変化に基づいて決定が行われます。これは通常、位相の切替えから競合が生じるためです。

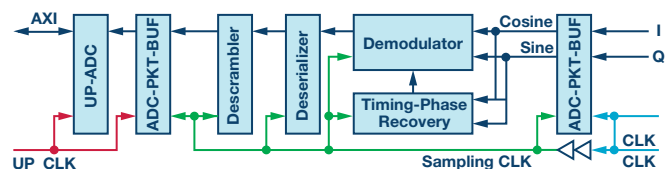


図6. 受信データ・パス

その後、復調されたビット・データがバイトに組み立てられてデスクランプリングされます。伝送バッファが空の場合、データは転送バッファに書き込まれます。CRCの検証はパケットの最後で行われます。CRCが一致して転送バッファへの書き込みが行われると、それがプロセッサ・インターフェースに通知されます。ソフトウェアにはこの要求をモニタすることが求められ、設定されている場合は、バッファの内容を読み出すことによってバッファを空にします。更に、その後もパケット転送を行うことができるように、要求をクリアする必要があります。

ビルド解説とダウンロード

本稿では、簡単なRFベースバンド・プロセッサの理論と実装の詳細を示し、この設計を実装したZC706およびAD-FMCOMMS3-EBZハードウェアについて解説しました。迅速なデモのために必要なすべてのものを備えた設計ファイルとビルド解説は、<https://wiki.analog.com/jp/resources/fpga/docs/hdl/xcomm2ip>で提供されています。wikiにも、HDL設計、ソフトウェア、RFセットアップ、性能および分析に関する詳細が示されています。

著者について

Rejeesh Kutty.

アナログ・デバイセズのカスタマ・ソリューション開発支援グループのHDLエンジニア。2011年入社。インド理科大学院にて電子工学の修士号を取得。

オンライン・サポート・コミュニティ



当社のオンライン・サポート・コミュニティで、アナログ・デバイセズの技術専門家と連携することができます。設計上の難問について問い合わせたり、FAQ を参照したり、話し合いに参加することができます。

ez.analog.com

* 英語版技術記事は [こちら](#) よりご覧いただけます。

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10F
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10F
名古屋営業所 〒451-6040 愛知県名古屋市西区牛島町6-1 名古屋ルーセントタワー40F

©2018 Analog Devices, Inc. All rights reserved.
本紙記載の商標および登録商標は、
各社の所有に属します。
Ahead of What's Possible は
アナログ・デバイセズの商標です。

www.analog.com/jp



想像を超える可能性を
AHEAD OF WHAT'S POSSIBLE™