

RM-007

超高速無線通信用ベースバンドシステム開発プラットフォーム A Baseband Development Platform for Ultra High-Speed Wireless Communications

大庭 信之 甲田 泰照 高野 光司 中野 大樹 山根 敏志 片山 泰尚
Nobuyuki Ohba Yasuteru Kohda Kohji Takano Daiju Nakano Toshiyuki Yamane
Yasunao Katayama

1. まえがき

近年、急激な情報化による、通信データ量増大、モバイル通信の拡大などから、無線通信の高速化、多機能化が望まれている。それらの流れから、大容量リアルタイム通信を可能とするミリ波通信等の次世代超高速無線通信が注目を集めている[1]。これまで、超高速無線通信を支えるさまざまな技術の研究開発が行われてきた[2]。しかし、今後の応用分野拡大、用途の多様化を見据えて、システムの高速化や低消費電力化、そして低コスト化などのさまざまな要求を満たしていかなければならない。われわれは、アンテナ、ベースバンド、そしてアプリケーションとのインターフェイスなどの多方面で研究開発を行ってきた [3, 4]。ミリ波を使ったギガビットクラス的高速伝送では、マルチパス、I/Q ローテーションなどの影響を受けやすく、理論解析、シミュレーションに加え、実動作環境での実験がシステム構築のため不可欠である。

今回、超高速無線通信システムの基幹技術のひとつであるベースバンドシステムに焦点をあて、ベースバンド開発プラットフォームを構築した。現在、その上で、MAC や PHY、アプリケーションインターフェイスの開発実装及び特性評価を行っている。大容量、高速、多機能で、かつ再構成可能な FPGA をベースに使うことで、多様なエラー訂正方式、変調方式、同期方式、フィルタ、プロトコルなどを実環境で実験、テスト、評価することができ、先に述べた高速化、低消費電力化など多様化する要求を満たす技術の開発がスムーズに行える。

本論文では、超高速無線通信用ベースバンド開発プラットフォームの基本構成及び特徴について述べ、続いて動画

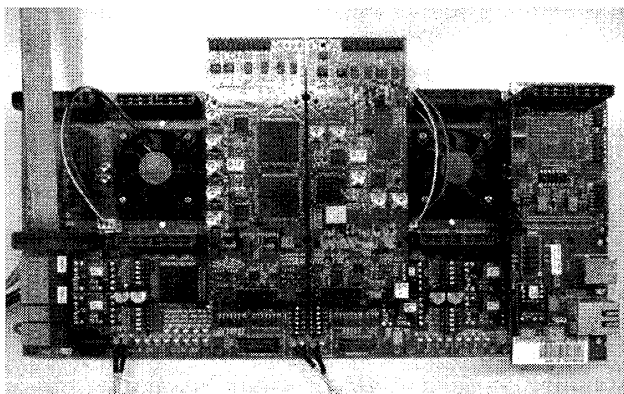


図1 ADC ドーターカード、DAC ドーターカード、USB インターフェイスカードが載った FPGA ベースボード

日本アイ・ビー・エム株式会社東京基礎研究所

IBM Research, Tokyo Research Laboratory, IBM Japan Ltd.

像伝送を実装評価した結果を報告する。

2. 開発プラットフォームの概要と設計ポイント

開発プラットフォームは大きく送信側と受信側に分かれ、表1に示す要素から構成される。

表1 プラットフォームの構成要素

送信側	FPGA ベースボード
	DAC ドーターカード
	送信 RF モジュール ビデオ入力カードなどのオプションカード
受信側	FPGA ベースボード
	ADC ドーターカード
	受信 RF モジュール ビデオ出力カードなどのオプションカード
システム制御 デバッグ用	USB インターフェイスカード

現在、FPGA ベースボードは2種類あり、それぞれ2個のFPGAと4個のFPGAを搭載する。2個のFPGAを使ったベースボード上にDACドーターカード、ADCドーターカード、USBインターフェイスカードを載せたシステムの写真を図1に示す。また、4個のFPGAを使ったベースボードとドーターカードの構成例を図2に示す。FPGAはXilinx社のVirtex-5 XC5VLX300を使用した。

今回開発プラットフォームを設計製作するにあたり、特

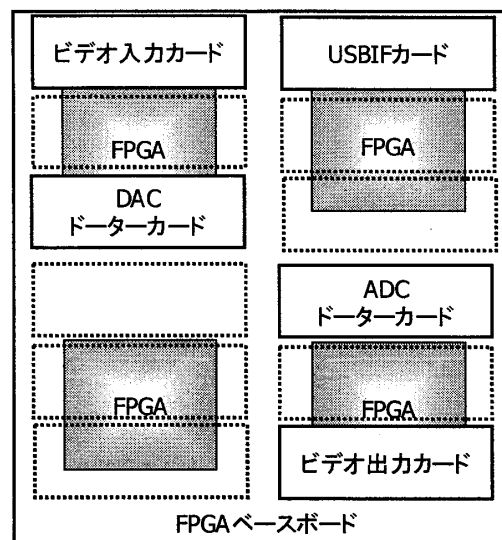


図2 4個のFPGAを使ったプラットフォームの構成例

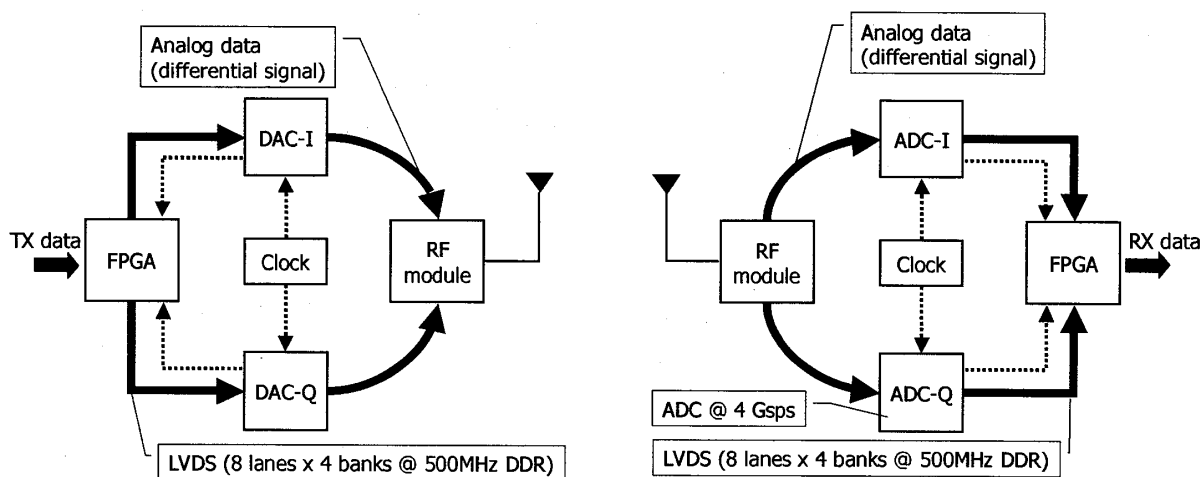


図3 送受信部のデータの流れとクロック構成

に、柔軟性と拡張性、そして高速信号のインテグリティに留意した。

図3に送信側と受信側のハードウェアブロック図を示す。ドーターカードはFPGAベースボード上にコネクタを介してマウントされる。ドーターカードはFPGA1個当たり最大3枚まで接続可能である。ここで、FPGAベースボードとドーターカードを結合するコネクタをすべて共通のものとして、配置や実装個数の制限を極力少なくしている。ドーターカードの電源は通常FPGAベースボードからコネクタを介して供給されるが、たくさんのドーターカードを搭載した際にFPGAベースボードからの総電力供給限度量を上回る場合に備えて、直接ドーターカードへ電源を別系統で供給することもできるように設計した。

ギガビットクラスの超高速無線通信ベースバンドシステムでは、FPGAをはじめとする構成モジュールの内部、及びモジュール間をつなぐバスに非常に高いバンド幅が要求される。本プラットフォームではFPGA内部のロジックが高速に動作すること、そしてFPGA間通信、FPGAとDAC間、FPGAとADC間をつなぐバスに高いバンド幅が求められる。

図3に示すように、送信データは、まずFPGAへ送り込まれ、誤り訂正、変調などの処理を行われた後、I相Q相のDACへ送られる。FPGAからDACへ送られるデータは、8レーンのLVDSをひとつのバンクとして、4バンク並列に500MHzクロックの両エッジに同期(DDR)して転送される。I相Q相それぞれ4バンク、合計8バンクである。

DACを駆動するクロック周波数は最高2GHzであり、そのときDACは毎秒4Gサンプルで動作する。DACから出たアナログ信号はRFモジュールへ同軸ケーブルを介して差動信号で送られる。DAC-IとDAC-Qに供給されるクロックは正確に同期している必要があるため、配線長、配線インピーダンス、マイクロストリップラインの引きまわしなどに細心の注意を払っている。また、本システムは実験研究を目的としているため、単に高いサンプリング周波数を目指すだけでなく、広い周波数範囲で動作できるように工夫した。具体的には、スプリッタなどのデバイスにつ

いては挿入損失を考慮した上でなるべく受動素子を使うようにしている。クロックのジッタはDACあるいはADCの有効ビット数(ENOB)に影響を与えるが、今回、クロック用として、広い周波数範囲で動作する小型低消費電力低ジッタ特性を持つ能動素子は入手できなかった。そのため、クロック源はDAC、ADCドーターカードには直接搭載せず、外部から同軸ケーブルで供給し、カード上でスプリッタによりI相とQ相へ分配、さらにバランを使ってシングルエンドから差動への変換を行っている。I/Q相のタイミングがずれないように、配線長などに注意を払った。

受信側は、まずRFモジュールからI/Q相のアナログデータがDACと同じように同軸ケーブルを介し差動信号でADCへ入力される。I相Q相それぞれのADCは最高毎秒4Gサンプルで動作し、AD変換したデータを1バンクあたり8レーンのLVDS×4バンクでFPGAへ転送する。I/Qでは合計8バンクとなる。DACと同じようにADCを駆動するクロックも細心の注意を払って設計した。

もともとFPGAはロジックの書き換えやピン配置など、ユーザがかなり自由にデザインできるものではあるが、LVDSをはじめとした高速I/O信号を扱う場合は、差動ペアとなる信号やクロック信号のピン配置は制約があり、ピン配置と配線の引き回しについては、あらかじめ十分検討しておかなければならない。特に今回は多数のLVDS信号を低スキューで動作させなければならないため、ピン配置とともに配線長や基板の配線層配置、ビアの配置にも気を配った。すべてのLVDS配線長の差は2mm以内に収めている。

電源系については、FPGAコア、FPGA入出力、DAC、ADC、クロックドライバなど異なった数種類の電圧を必要とするため、消費電力を考慮しながら、低ノイズのLDO(低ドロップアウトリニア電圧レギュレータ)を中心に構成した。

一方、DAC、ADCのセットアップ、RFモジュールのコンフィギュレーションなど速度や消費電力に比較的余裕のある部分は、入手しやすい部品を採用し、低コスト化を図っている。

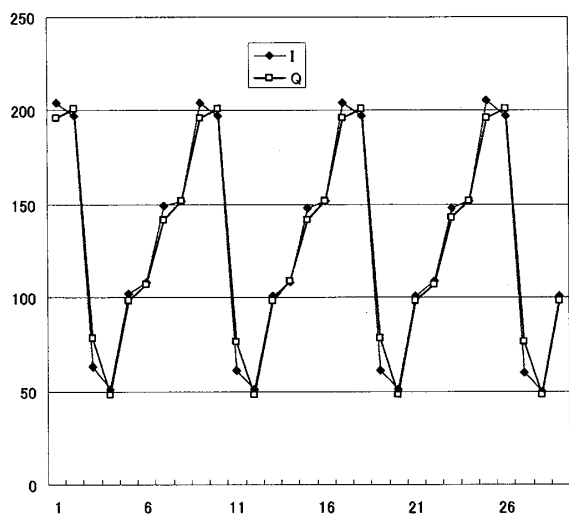


図4 I/Q波形の観測

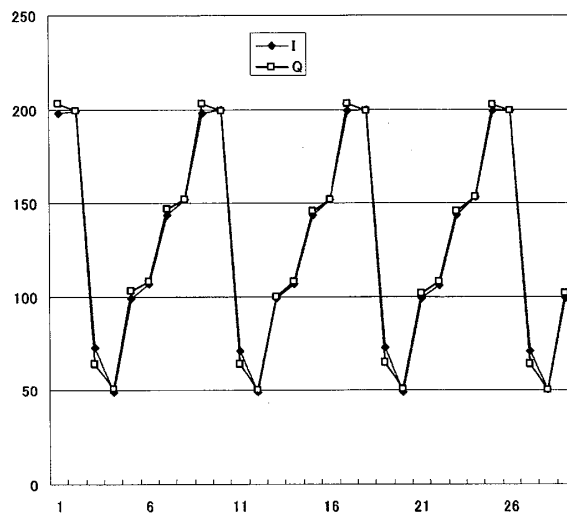


図5 I/Q波形の観測、I/Q逆接続

システム制御については、PCからFPGAベースボード上に載ったUSBインターフェイスカードを通して行う。さまざまな実験を自由に行えるように、USBターゲットマイコンのファームウェア、PC上で動作するのデバッグについても独自開発している。システム制御はFPGA内部に実装した制御レジスタを読み書きすることで行い、たとえば制御レジスタを書き換えることでDACやADCの制御線をオン・オフする。また、ADCに用意されている位相、オフセット、ゲインの調整機能はFPGAとADC間に張られた独自のカスタムシリアルバスを通して制御する。

RFモジュールは弊社ワトソン研究所で開発されたものを使っている[2]。

3. 基本動作テスト

本プラットフォームの動作テストを次のように行った。

ベースバンド部分の動作を把握するため、RFモジュールは使用せず、DACのアナログ出力を直接ADCのアナログ入力と接続した。I相とQ相を別々に同軸ケーブルによって接続し、送信側FPGAからいろいろなパターンデータを送り、それらを受信側FPGAで観測した。

図4は16QAM伝送を想定して、I、Qそれぞれ同じ4レベルの階段状波形を繰り返し送って測定した例である。サンプリング周波数は3GHzである。IとQのデータが振幅、位相ともよく一致していることが分かる。同軸ケーブルの長さの影響を含むI/Qの振幅ずれ位相ずれを確認するため、IとQの同軸ケーブルを挿し替え、再び同じデータで観測を行った。その結果を図5に示す。この場合もI相Q相の振幅、位相ともよく一致していることが分かる。高速クロックの分配、アナログデータ配線などのIQバランスが良く取れていることが確認できた。

4. 実動作テストと評価

続いて、1600x1200ピクセルのUXGA動画データを用いて動作テストを行った。基本動作テストと同じく、DACの出力を直接同軸ケーブルを介してADCへ送り込む。実験風景を図6に示す。FPGAに動画用の簡単なMACとIEEE 802.15.3準拠のPHYを実装した[5]。データの流れを

図7に示す。DACとADCのサンプリング周波数は3.5GHzである。変調方式としてQPSKを使用し、UXGA画像データ転送レート2.86Gbpsを達成した。

同期回路についてはデータレートの2倍のオーバーサンプリングを行ったあと、デマルチプレクサーでEvenとOddに分配する。IEEE802.15.3cで定義されたプリアンプルは、ゴレイ符号で構成されているので、ゴレイ相関器によって相関値を計算していく。相関値のピークを検出することで、プリアンプルの検知が行われる。検知に成功すると、シンボル同期の位置、IQ回転の量、そしてEvenとOddの相関値比が得られる。シンボル同期の位置情報はバレルシフタのシフト量を決める。IQ回転の量からは、その複素共役を乗ずることで回転を戻すことができる。そして、相関値比を利用して、EvenとOddの補間を行う。こうすることで、2倍でオーバーサンプリングされたデータから、

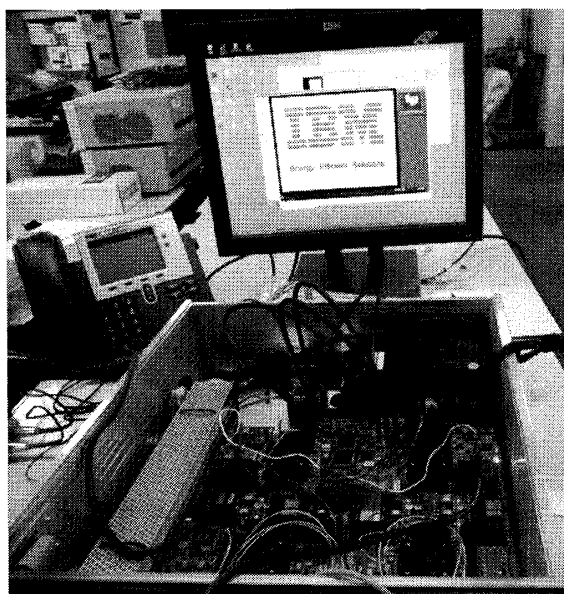


図6 実動作テスト

元のデータレートへ戻している。

5. まとめ

1Gbps を超える超高速無線通信用ベースバンド開発プラットフォームの構成、特徴、実装評価について述べた。現在、RF モジュールと接続し、無線を使った本格的な実験を始めている。今後 FPGA の特性と本プラットフォームの柔軟性を生かし、さまざまなアルゴリズムやプロトコルを実環境で実装、テスト、評価していく予定である。また、ビームフォーミングなどの複数のアンテナを用いた実験も行っていく予定である。

参考文献

- [1] 総務省, “ワイヤレスブロードバンド推進研究会最終報告書,” 2005.
- [2] B. Floyd, S. Reynolds, U. Pfeiffer, T. Beukema, J. Grzyb, C. Haymes, “A Silicon 60-GHz Receiver and Transmitter Chipset for Broadband Communications,” International Solid State Circuit Conference (ISSCC) Dig. Tech Papers, pp. 184-185, Feb. 2006.
- [3] Y. Katayama, et al., “2-Gbps Uncompressed HDTV Transmission over 60-GHz SiGe Radio Link,” IEEE Consumer Communications and Networking Conference (CCNC), Jan. 2007.
- [4] U. Pfeiffer, B. Floyd, S. Reynolds, A. V. Garcia, C. Haymes, Y. Katayama, D. Nakano, T. Beukema, and B. Gaucher, “Silicon Millimeter-Wave Radio Circuits at 60-100GHz,” Topical Meeting on Silicon Monolithic Integrated Circuits in RF Systems (SiRF), Jan. 2007.
- [5] IEEE Computer Society, “802.15.3 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs),” IEEE Standards, 2003.

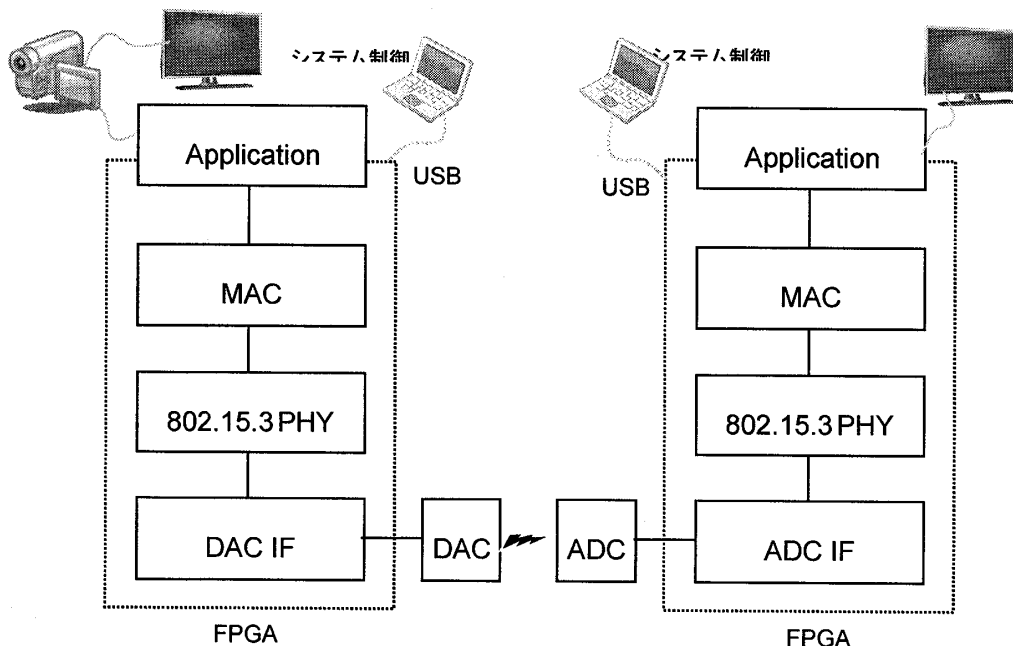


図7 UXGA 画像ベースバンド伝送の機能ブロック構成

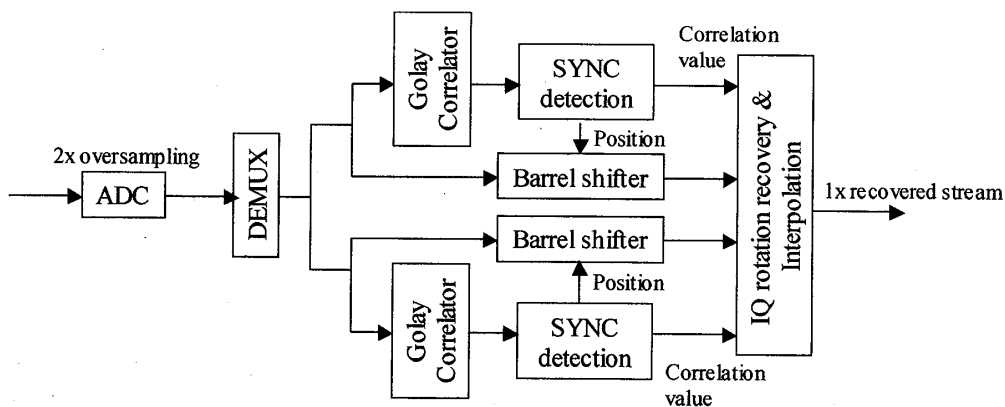


図8 ベースバンド通信実験用 PHY 内部の同期検出調整器の構成