

RFSoc の MTCA 規格制御カードへの実装 IMPLEMENTATION TO RFSoc MTCA CONTROL CARD

漁師雅次[#], 岩城孝志, 出口久城, 濱洲竜斗, 林和孝, 張替豊旗

Masatsugu Ryoshi[#], Takashi Iwaki, Hisakuni Deguchi, Ryuto Hamasu, Kazutaka Hayashi, Toyoki Harigae
Mitsubishi Electric TOKKI Systems

Abstract

In accelerator control, it is important to monitor and control the amplitude and phase of high frequency signals to create a stable beam. Fast digital processing is essential for flexible and robust controls. A modern RFSoc (radio frequency systems on the chip) enable digital conversion of RF analog signals up to the S-band to A/D (analog/digital) or D/A (digital/analog) directly. The digital signals are processed by FPGA (field programmable gate array) and CPU in the RFSoc. This eliminates analog circuits with nonlinear characteristics. Using this RFSoc, we have developed a standard MTCA (Micro Telecommunication Computing Architecture) control card with a small board size of 73.5mm × 180.6mm. In addition, EPICS IOC is embedded into the APU (application processor unit) and the RPU (real-time processor unit) for accelerator users.

1. はじめに

高周波加速空洞の電圧強度と位相を安定させるために LLRF (Low Level RF) は、クライストロンの出力や空洞の入出力の高周波信号をモニタしてフィードバック・フィードフォワード制御などを行う。また、加速ビーム軌道を把握するために、BPM (Beam Position Monitor) システムにより、ビーム信号から取得した高周波信号を狭帯域でフィルタリングして加速空洞内のビーム位置を算出する。従来は、これらの高周波信号を検出して信号処理するためにアナログの検波回路を用いていたが、近年ではアナログ回路の周波数コンバータを使い、十数 [MHz] の中間周波数に変換した信号を、その 4 倍や 8 倍などのサンプリング周波数で A/D 変換してデジタル化し、FPGA や DSP (Digital Signal Processor) などを用いてデジタル信号処理を行っている [1]。さらに、500 [MHz] 程度までの RF 信号を直接 A/D 変換して信号処理を行う機器も出てきた [2]。

ここにきて 6 [GHz] までの RF 信号を直接 A/D 変換できる RFSoc が Xilinx からリリースされている。このデバイスは、C バンドまでの加速器の LLRF や BPM などの RF 信号を直接デジタル化して制御するシステムへの適用が期待されている。2019 年には、Xilinx 製の RFSoc の評価ボード ZCU111 を使った 19 インチラックマウント可能な試作ユニットで性能評価を行った [3]。SNR およびクロストークならびに振幅・位相安定度の評価を行った。その結果から、信号処理方法を工夫して、加速器制御に利用できる可能性が示唆された。

そこで加速器制御では標準的なプラットフォームになってきた MTCA 規格の試作カードを開発して、高密度実装技術をつかい小型化して多チャンネル化ができるプラットフォームを目指す。これにより、複数の 19 インチラックマウントユニットで構成していたシステムがより小型化できると考えられる。

2. MTCA 規格制御カードの構成

2.1 MTCA 規格の特徴

通信機器用のプラットフォーム規格である ATCA (Advanced Telecom Computing Architecture) 規格の子基板の規格である AMC (Advanced Mezzanine Card) 規格がある。MTCA 規格は、この AMC をシェルフ (ラック・クレートと同様) に直接実装して使うシステムの規格である。ATCA と同様にシェルフの状態を管理するためのシェルフマネージャの機能を持った MCH がある。主な構成要素を以降に述べる。

MTCA シェルフには、AMC がささるバックプレーンがあり、高速シリアル通信するための P2P の配線やタイミング信号等を共有するためのバス配線がある。また、シェルフマネージャとしての MCH (MTCA Carrier Hub) と通信するための IPMI (Intelligent Platform Management Interface) の配線がある。AMC などを冷却するための CU (Cooling Unit) があり、ホットスワップが可能である。また、FAN の回転速度は MCH によりシェルフ内温度に応じて制御される。各カードへの電源供給は MCH に制御される PM (Power Module) からバックプレーン経由で行われる。

MCH は、PM・CU・AMC の管理および高速シリアル通信のための中心となり、スイッチとなっている。冗長系の MCH と合わせてデュアルスター接続される。

電源は、メイン回路動作用の PP (Payload Power) が 1 スロットあたり +12 [V] 80 [W] まで使える。これとは別に IPMI 回路を動かすための MP (Management Power) として 1 スロットあたり +3.3 [V] が 150 [mA] まで使える。MP は、スロットに AMC が実装されると認識したら供給される。これにより AMC 上の IPMC が起動して、MCH と通信をはじめて FRU (Field Replaceable Unit) 情報を通知して、MCH はシェルフ内の状態を確認して、AMC に PP を供給しても大丈夫ならば PM に対して該当スロットの PP を供給するように指示を出す。この電源供給の仕組みを使い AMC のホットスワップを実現している。

[#] ma-ryoshi@west.melcos.co.jp

2.2 試作カードの構成

試作カードの特徴を Table 1 に示した。主要構成部品は、RFSoc と DDR4-SDRAM と POL (Point Of Load, 電源モジュール) となる。RFSoc には、PL (Programable Logic, FPGA 部分) と PS (Processing System, CPU 部分) があり、電源が起動すると PS に接続された SD カード内のブートローダが PS で起動して、FPGA コンフィグレーション情報を展開して PL が起動する。同時に PS 用の u-boot がロードされ Linux が起動する。Linux を動作させるために PS にはワーキングメモリとして DDR4-SDRAM が 4[GB] 接続されている。別途、PL 部にも DDR4-SDRAM が 4[GB] 接続されており、ADC 結果やデジタル演算中のデータ収集用のメモリとして使用している。

Table 1: Specifications of Prototype Card

FPGA	XCZU27DR-1FFVG1517I
PL	Logic Cell: 930K BRAM: 38.0Mb UltraRAM: 22.5Mb DSP Slices: 4272 GTY Transceivers: 16
PS	Quad-core Arm Cortex-A53 Dual-core Arm Cortex-R5F
OS/Software	Ubuntu Linux 16.04 EPICS Base 3.14.12.3, Sequencer, Autosave
RAM	DDR4-3200 4GiB×2 (PL, PS)
FPGA Configuration	Micro-SD Card, Remote Update
ADC	8ch, 12bit, 4.096GSPS max., BW 4GHz
DAC	8ch, 14bit, 6.556GSPS max., BW 4GHz
Zone1 (AMC Connector)	Port [0:1]: 1000BASE-BX, Port [2:3]: Storage Port [4:11]: PCI Express Gen3 Port [12:15]: P2P Port [17:20]: M-LVDS FCLKA TCLKA~D IPMB: IPMI v1.5 support
Interface	Trigger IN/Clock IN Trigger OUT/Clock OUT
Switch	8bit DIP-switch
Front Panel LED	Blue: Hot swap status Red: Error status Green: Running status
Size	PCIMG MTCA.0 Single-Width Full Size 73.8*28.95*181.5 [mm]

カードのフットプリントを極力小型にするために、シングル幅とした。そのため、前面のパネル面のサイズが

73.8[mm]×28.95[mm]と狭くなる。使用する RFSoc は高速 ADC および DAC が各 8ch の入出力あるため合計 16ch の同軸インタフェースが必要となる。信号モニタ機能として使う用途が多いと想定し、ADC に接続した RF 入力を RFSoc が実装される親基板に実装した。ここで使うコネクタは、Super KEKB 等の LLRF でも使ってきた Nicomatic 社製の多極同軸コネクタを選定した。カードは Fig. 1 のように 3 段構成として 2 段目に将来拡張用のデジタルインタフェースを実装できるようにして、3 段目に DAC 出力用の同軸コネクタを実装した。

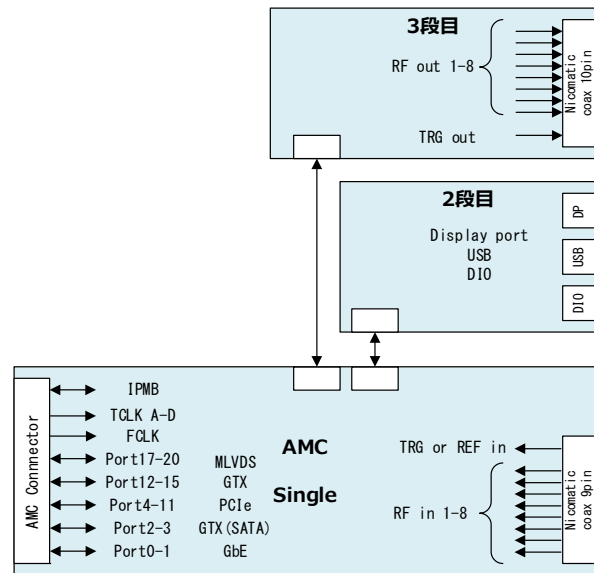


Figure 1: Card configuration.

バックプレーン側のカードエッジコネクタは、一般的なデュアルスター型のバックプレートポロジで使うために、Port0・1 のベースインタフェースの GbE、Port2~3 はストレージ用のインタフェースに拡張できるように高速シリアル I/O を接続した。Port4~11 には PCIe を実装できるように接続して、Port12~14 にはカード間 P2P 接続可能なようにした。Port17~20 は各スロットで信号共有できるバス接続の MLVDS 入出力とした。MCH から各スロットに供給されるクロックは、通信用の低速クロック TCLK A~D (~50[MHz]) と PCIe 用の FCLK (100[MHz]) クロックを接続した。また、前面コネクタからもクロックおよびトリガ入力ができるようにした。

RFSoc を実装した試作カードの外観を Fig. 2 に示した。パネルサイズは横方向にシングル幅もしくはダブル幅の 2 種類があり縦方向に、ショートサイズ・ミッドサイズ・フルサイズの 3 種類がある。一般的には、MTCA ではミッドサイズが多く、冷却能力の向上 (伝導冷却ブロックなど) を考慮に入れてフルサイズにする場合がある。MTCA.4 規格など RTM (Rear Transition Module) を用いる場合は、ダブル幅にしてフロントバックプレーン上部の ZONE3 というエリアを使ってコネクタ接続して RTM と通信できるようにする。今回は、フットプリントを最小にすることおよび RTM を使わないことから、シングル幅にした。パネルの設計は、使用するコネクタの型名およびコネクタの位置が決まれば、3D-CAD を使い干渉が無いかなどを確認

する。レバー部分及び LED 用ライトパイプの取り付け金具は既製品を使い、パネル部分のアルミプレートおよび文字表示シートのみカードごとに新規で設計する。



Figure 2: Photo of prototype card.

3. 試作カードでの工夫

3.1 フロアプラン

MTCA.0 規格シングル幅フルサイズのカードサイズは Fig. 3 のような $73.8[\text{mm}] \times 181.5[\text{mm}]$ である。プリント基板のサイズが狭く、高密度実装になったため、パターン設計を進めるとき従来とは異なる工夫をした。

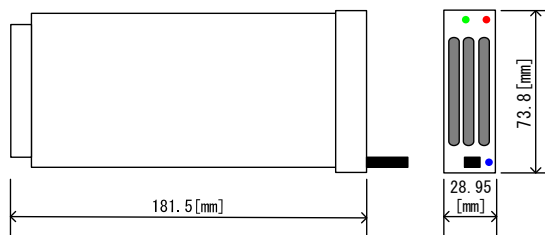


Figure 3: MTCA single-width full-size.

RFSoc のサイズが $40[\text{mm}] \times 40[\text{mm}]$ であり、多種の電源電圧が必要であり、POL が複数必要になった。また、PS の CPU を使うための DRAM が必須であり、高密度の実装が必要となった。そこで、プリント基板のパターン設計を始める際に、部品間の信号や電源の接続および多層基板の層構成を概要検討した。このフロアプランをしておき、パターン設計の後半での変更による大幅な手戻りを減らしリスクを軽減した。

試作カードでは、前面側に RF 信号入出力、背面側に高速シリアル信号入出力を接続する必要があった。また、PS の CPU 用の DRAM を接続、また PL のロジック用の DRAM を接続した。まず、外部との接続がある RF 信号および高速シリアル信号の RFSoc のピン配置からプリント基板上での設置角度を決めた。次に、高速のデジタル信号で配線設計に注意が必要な DRAM をどこに配置するかを決めた。1 つの DRAM に接続するアドレスおよびデータなどで約 40 本をどのように配線するかをの経路を

決めた。特に、信号相互のクロスカップリングや電源供給について計画し、ラインアンドスペースを設計ルールに従いパターン設計した。また、RFSoc には、RF 回路・PL 部・PS 部・高速シリアル部の回路ブロックごとに複数の電源があり、必要な電流容量に適したパターン幅にして信号線同様クロスカップリングに注意して配線した。信号線および電源パターンともにベタ GND パターンと隣接層でペアの配線にしてリターン経路を確保して、ノイズ放射の低減を配慮した。信号種ごとにおおよその経路を各層のどこを通すか事前に計画して、上のルールの適用漏れをなくせた。

3.2 モデルベース開発

RFSoc では、多チャンネルの ADC, DAC からなる、信号処理を取り扱うため、デジタルフィルタや数値演算処理の規模がより大きくなる傾向にある。従来から FPGA を使った信号処理の設計では、評価済みの FPGA IP (Intellectual Property) ライブラリを使った、HDL (Hardware Description Language) のハンドコーディングである。システムの要求仕様からブレイクダウンする際に、誤認識による不具合が HDL のハンドコーディングにより混入する恐れがある。また、設計検証では、別途用意した HDL で記述した入力ベクタモデルを使い、出力を期待値一致の確認もしくはロジックアナライザのようなデジタル波形を目視で確認している。HDL シミュレータではアナログ信号のシミュレーション用ベクタファイルを作りにくかったため、あまり多くの検証パターンを流せず、代表的なパターンだけ流している。

そこで、数値演算ツールの MATLAB/Simulink を使った機能モデルを作成してシステム動作の検証と要求機能のモデルを作成する。このモデルの機能の一部を FPGA に実装する際には、オプションの HDLCoder により直接 HDL へ変換する。モデルベースの設計と自動コーディングにより、不具合の混入を防げる。また、MATLAB/Simulink を使いシステムレベルのモデルを作って、周辺システムを含めたシステムレベルのシミュレーションをするので、処理パラメータ調整と実機への実装が容易にできるため、開発サイクルの短縮できる。また、ライブラリから容易に検証の準備ができ、シミュレーション結果も、時間軸や周波数軸など豊富なグラフ化機能を使ってわかりやすく表示ができ結果の確認も容易である。また、シミュレーション時間も HDL シミュレーションより早く、時間が短縮でき効率的にできるようになった。

モデルベース開発の課題は、HDLCoder で出力した HDL は可読性が悪い。また、デジタルフィルタなどは IP ライブラリ利用やハンドコーディング時よりも合成後の使用リソースが多くなる傾向がある。今後の改善に期待したい。

3.3 Python を使ったプロトタイプ

PS 上の Linux で動作するプログラム言語 Python から PL 上のレジスタやメモリにアクセスできるライブラリを開発し、A/D 変換したデータの収集および D/A 変換へ出力するデータの生成ができるようにした。これにより、実機のアナログ入出力をつかって、Python による様々なライブラリ利用やアルゴリズム開発と動作検証ができる。但し、リアルタイムデータ処理ができるわけではないので、

間欠的にある程度のサンプリング数のデータを処理するシステムへのプロトタイプへの適用に限られる。

3.4 クロック生成回路

複数の AMC で動作周波数の同期を取るためには、一般的には、基準クロックを別途分配する必要がある。MTCA シェルフ内の AMC では、バックプレーン経由にて、PCIe 用の FCLK、あるいは、MCH の前面から入力される TCLK を使い、各 AMC へ分配し同期させる事ができる。

今回、外部からクロックを入力しなくても複数の AMC の動作周波数を同期させる、Fig. 4 のような構成を加えた。これは、SYNC-E(SYNChronous Ethernet,)と同様の方法を使って、高速シリアル通信の受信データからクロックデータリカバリーされたクロックを基準信号として PLL を使い、このカードで必要なクロックを生成する。これにより、遠隔のカードの動作クロック周波数に同期させられる。これにより基準クロックを別途分配して供給する必要がなくなる。

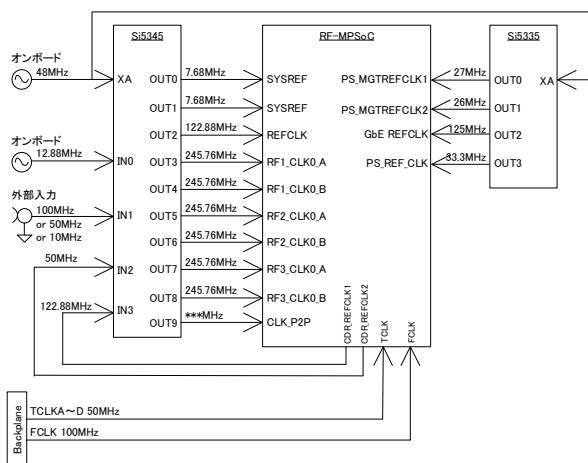


Figure 4: Clock generate circuit.

今回の AMC で装備する、8b/10b を用いた高速シリアル通信は、送信側でデータ 8 ビットを 10 ビットのシリアルデータに変換する際にクロック情報も組み込み伝送できる。受信側では、送信側の周波数とほぼ同じ周波数の基準クロックを使い、同期用のデータ(一般的には K28.5)というコードを受けて PLL にてデータ受信クロックを再生する。この再生クロックを外部 PLL の基準クロックに切り替えて入力して、送信側と受信側の周波数を完全に同期させられる。但し、外部 PLL の位相差検出用の分周クロックと出力クロックの周波数の関係から決まる位相の取り得るパターンに応じて、クロック再生動作ごとに位相が不確定になる。そのため、送信および受信回路でクロックの周波数だけでなく、位相の同期が必要な用途には使う場合はさらに回路上の工夫が必要と考えられる。

3.5 EPICS IOC の実装

PS 内の Linux に EPICS IOC を実装して PL 部のレジスタやメモリを制御できるようにした。既存の制御カードの PPC(Power PC)内蔵 FPGA や Zynq SoC を使った EPICS IOC と同様に加速器システムでユーザが利用し

やすいようにした。

まず、従来から実績のある EPICS base 3.14.12.3 を実装して、Sequencer および Autosave も従来通り組み込んだ。PL 部へアクセスドライバは従来から使っているドライバ・デバイスサポートを流用した[4]。RFSoc の PS 部(CPU)は Cortex-A53 であり、パフォーマンスが向上したため、組み込み Linux の制約も大いに緩和され、ソフトウェア開発にはクロスコンパイルではなく、セルフコンパイルが利用でき開発環境の構築が容易となった。

4. まとめ

RFSoc を MTCA.0 シングル幅フルサイズの AMC に実装した試作カードを開発した。RF 入出力各 8ch が可能となり、加速器制御やモニタ回路として、C バンドの一部までの高周波信号を直接取り扱う事を想定して。現在、試作カードの組み立てが完了して、単体の性能評価を進めている。小型高密度実装をしたためいくつかの課題が見えてきており、原因探求と対策を進めている。

今後、詳細な性能評価をすすめ、LLRF や BPM などの信号処理を実装して、実システムに適用できるかどうかの評価を進めていく予定である。また、本 AMC 単体を小型のモジュールとして使えるケースを開発して、適用できる用途を広げていく。

参考文献

- [1] T. Kobayashi *et al.*, "Operation status of LLRF control system for SuperKEKB Commissioning" Proceedings of the 13th Annual Meeting of Particle Accelerator Society of Japan, Chiba, Aug. 2016, pp. 836-840.
- [2] T. Ohshima *et al.*, "Upgrade of LLRF system at SPring-8 storage ring using MTCA.4 standard modules" Proceedings of the 15th Annual Meeting of Particle Accelerator Society of Japan, Nagaoka, Aug. 2018, pp. 55-59.
- [3] M. Ryoshi *et al.*, "Application study of MP-RFSoc to accelerator control" Proceedings of the 17th Annual Meeting of Particle Accelerator Society of Japan, Online, Sep. 2020, pp. 555-559.
- [4] H. Deguchi *et al.*, "Linux and EPICS embedding on MTCA.4 FPGA(ZYNQ) board" Proceedings of the 11th Annual Meeting of Particle Accelerator Society of Japan, Aomori, Sep. 2014, pp. 1315-1319.