MTCA における PCIe DMA と 40GbE を組み合わせた高速大容量データ伝送

HIGH-BANDWIDTH DATA HANDLING USING PCIE DMA AND 40-GBE WITH MTCA

漁師 雅次#, A), 岩城 孝志 A), 越智 圭一 A), 林 和孝 A), 張替 豊旗 A), 平田 雄一 A), 山崎 伸一 A)
Masatsugu Ryoshi #, A), Takashi Iwaki A), Keiichi Ochi A), Kazutaka Hayashi A),
Toyoki Harigae A), Yuichi Hirata A), Shinichi Yamazaki A)
A) Mitsubishi Electric TOKKI Systems Corporation

Abstract

Various sensors information is used in the operation of the accelerator. Because the bandwidth of the transmission line is narrow compared to the acquisition data bandwidth, it reduces the amount of data to be transmitted and shares it by only thinning out or judgment result. Therefore, analysis of a peculiar phenomenon could not be performed in some cases. Computer environment that can collect data from multiple sensors including IoT (internet of Things) in recent years and process it, process it as raw data, transmit it as it is, and do advanced processing relatively easily It has become possible to do. We constructed a data acquisition system utilizing the high-speed serial transmission path on the backplane of MTCA (Micro Telecommunications Computing Architecture) we have been using for LLRF, BPM, and image acquisition and evaluated its performance.

1. はじめに

加速器制御では、加速空胴内の RF およびビームの状態をディジタル化して収集し処理している。加速ビームの高いエネルギー精度や安定度を得るために、各情報の高精度化・高時間分解能化が必要となり、取り扱われる単位時間あたりのデータが大容量化してきている。そのため、従来からの制御システムで使われている VME バス(VME64:40[MB/sec] max.)や Compact PCI バス(64bit/66MHz:533[MB/sec] max.)よりも広帯域な伝送方法が求められつつある。その対策の一つとして標準規格とは異なる方式、例えば Xilinx 製 FPGA で使える高速シリアル伝送プロトコルの Aurora を使いデータ伝送の広帯域化を実現する方法がある。この方法は、専用ハードウェア間で接続する場合は比較的容易に実現できるため有効であると考えられる。しかし、データを集約して演算するために汎用のコンピュータへデータを渡す際、カスタムのハードウェアを準備する必要があるため現実的ではない。そこで、汎用コンピュータとの親和性のよい Ethernet を用いた高速データ伝送を、加速器制御で広まりつつある MTCA をベースに構築して評価した。[1]

まず、MTCA のバックプレーンを使った高速伝送評価システムの構築を行った。MTCA のバックプレーンには、GbE(Giga-bit Ethernet)を基本通信手段として、さらに fat pipe という高速差動伝送路が複数レーンある。基本的な通信経路のトポロジーは MCH を中心とするスター配線となっている。つまり、MCH(Micro TCA Carrier Hub)に fat pipe で使用するプロトコルのスイッチ機能が実装されている必要がある。このプロトコルは、PC で広く使われている PCIe Gen3(PCI Express Generation3)を採用されていることが多い。これは、CPU およびチップセットとの親和性が高いためだと考えている。そこで、PCIe で通信できる AMC(Advanced Mezzanine Card)を開発して、理論伝送速度の約 8 割の性能がでることを確認した。

次に、Ethernet の中で広帯域かつ比較的入手性の良くなってきた 40GbE を用いたシステム構築のための手法を確立するために、Intel が策定した DPDK(Data Plane Development Kit)を用いて伝送性能の評価をした。この結果、フレーム生成の簡易的な評価用ソフトで 40GbE (40 Giga-bit Ethernet)の理論伝送速度の約 9 割の性能がでることを確認できた。但し、使用するコンピュータのメモリ構成および CPU 性能が伝送速度へ影響することが分かった。この評価において、MTCA で使用している CPU-AMC のメモリの帯域幅がボトルネックとなり、40GbE および PCIe の伝送性能を最大限に引き出すことができないことが分かった。

今回は、実際の加速器制御や DAQ でも使用できるように、MTCA に実装した高速 ADC ボードから 40GbE の NIC(Network Interface Card)経由で外部の WS(WorkStation)へ伝送し、HPC(High-Performance Computing)で使われる階層化データ形式である HDF5 (Hierarchical Data Format 5)へ変換するシステムを構築して性能を評価した。

2. 評価システムの構成

40GbE の評価に使用した MTCA シェルフを Figure 1 に示す。

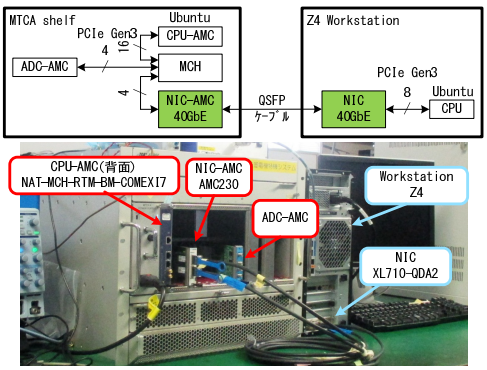

Figure 1: Evaluation test environment of 40GbE.

ma-ryoshi@west.melos.co.jp

MTCA のバックプレーンは MCH を中心として各 AMC スロットがスター接続されている。Port0 がプライマリ MCH、Port1 がセカンダリーMCH（冗長化システム用）に接続されており GbE の通信が可能である。また、Port4～7 の fat pipe はプライマリ MCH に接続されており、Port8～11 はセカンダリーMCH に接続されており、これらも冗長化に対応した接続になっている。今回は、PCIe Gen3 のスイッチを搭載している NAT 製 MCH により、Port4～7 を PCIe の 4lane として利用する。

MTCA シェルフ内の構成は CPU-AMC を「AMC754-002-100-000」に更新した。従来使っていた CPU カードと性能比較を Table 1 に示す。CPU が「Core-i7 3517UE」から「Xeon D-1548」に変わり、メモリ帯域が 68[Gbps]から 137[Gbps]に向上されるため 40GbE および PCIe の DMA の伝送速度の改善を見込んだ。

Table 1: Equipment Configuration of MTCA CPU Card

	NAT-MCH-RTM-BM-COMEXI7	AMC754-002-100-000
CPU	Core-i7 3517UE 2Cores,4threads 1.7GHz	Xeon D-1548 8Cores,16threads 2GHz
Memory	4GB ECC DDR3-1066 64[Gbps]	16GB ECC DDR4-2133 137[Gbps]
OS	Ubuntu 14.04.4 LTS Kernel 4.2.0-27	Ubuntu 18.04 LTS Kernel 4.15.0-23

また、WS を「HP 製 Z4 G4 Workstation」に更新した。従来使っていた WS との性能比較を Table 2 に示す。メモリの帯域幅が 102[Gbps]から 171[Gbps]に向上しており PCIe と 40GbE の理論伝送速度に比較して余裕ができた。また、データの保存先に M.2 の SSD を実装したが、まずは RAM ディスクで評価した。

しかし、MTCA の CPU カード「AMC754」のセットアップが進まず、MTCA 側は従来と同じ構成で動作確認することになった。

Table 2: Equipment Configuration of Workstation

	Z820	Z4
CPU	Xeon E5-2643 4Cores,8threads x2CPUs,3.30GHz	Xeon W-2125 4Cores,8threads x1CPUs,4.00GHz
Memory	64GB ECC DDR3-1600 102[Gbps]	64GB ECC DDR4-2666 171[Gbps]
OS	CentOS 7.2 Kernel 3.10.0-327	Ubuntu 18.04 LTS Kernel 4.15-23
NIC	XL710-QDA2 40GbE	←

3. MTCA ベースの 40GbE のデータ伝送評価

MTCA ベースのデータ収集システムにおいて、上位装置へ 40GbE を使ったデータ伝送できるソフトを新規に開発した。Figure 2 のようなシステム構成で PCIe DMA と 40GbE を組み合わせたデータ伝送の評価をした。

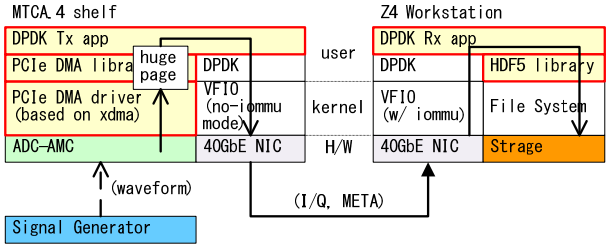

Figure 2: Software layers on test environment.

3.1 PCIe Gen3 インタフェースの高速 ADC の AMC

最高 4GSPS の高速サンプリングが可能な ADC を搭載した AMC タイプの FPGA（Field Programmable Gate Array）ボードを開発した。処理系統図を Figure 3 に示した。オンボードメモリに DDR4 SDRAM を 2[GB]搭載しており、大容量データをバッファリングすることができる。また、バッファリングされたデータは PCIe Gen3 インタフェースにて高速データ転送が可能である。FPGA は Xilinx 社製 Kintex UltraScale（xcku040-ffva1156-2-e）を採用しており高度な信号処理が可能である。

高速 ADC で量子化された RF 信号（アンダーサンプリングされた IF 信号）は FPGA にてディジタル処理される。FPGA で DDC（Digital Down Convert）および FFT（Fast Fourier Transform）処理する。リアルタイムに入力信号のスペクトラムの第 1 から第 5 までのピーク値と SNR（Signal to Noise Ratio）を抽出する。

DDC 後の IQ データは 64[MB]区切りで 33[msec]毎にオンボードメモリを使用したリングバッファ(8 面)にリアルタイム保存される。メモリコントローラと PCIe Endpoint を AXI4（Advanced eXtensible Interface 4）バスで接続する事で大容量 IQ データの高速転送を可能としている。

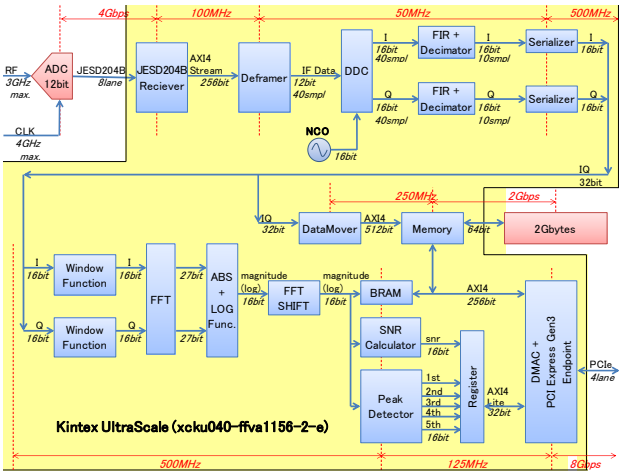

Figure 3: Picture of 12bit-4GSPS A/D AMC with PCIe Gen3 4lanes.

3.2 PCIe DMA と DPDK のバッファ共有

ADC-AMC 上に収集した IQ データを PCIe の DMA 転送で CPU-AMC のメモリ上に確保されたバッファ領域へ転送する。その後、NIC-AMC（Network Interface Card）を経由して 40GbE で外部の WS へ伝送する。40GbE の回線帯域幅を高効率で使用するために、DPDK を使った独自のデータ伝送アプリケーションソフトを新たに開発した。
データ収集から伝送までの概要は次の通りである。

まず、ADC-AMC 上の DDR4-SDRAM に 64[MB]の IQ データを蓄積されると、PCIe 上に割り込みで通知する。CPU は、ADC カード上の FPGA に実装している DMA コアに Scatter Gather 用の転送元および転送先のアドレスリストを渡し DMA の開始指示をする。DMA が完了すると PCIe 上に割り込みで通知されるので、その後 DPDK を使って外部の WS へデータ伝送する。

DPDK は、データ伝送するためだけの技術であるため Ethernet のフレームデータへの変換は別途処理を用意する必要があった。処理のオーバヘッドを極力なくすために DMA 転送開始前に Figure 4 のようなフレームフォーマットを CPU-AMC のメモリ上に作成しておき、このデータペイロードに合うように IQ データを DMA 転送して格納する。これで、CPU とメモリ間のデータのやりとりを極力少なくすることが可能となる。

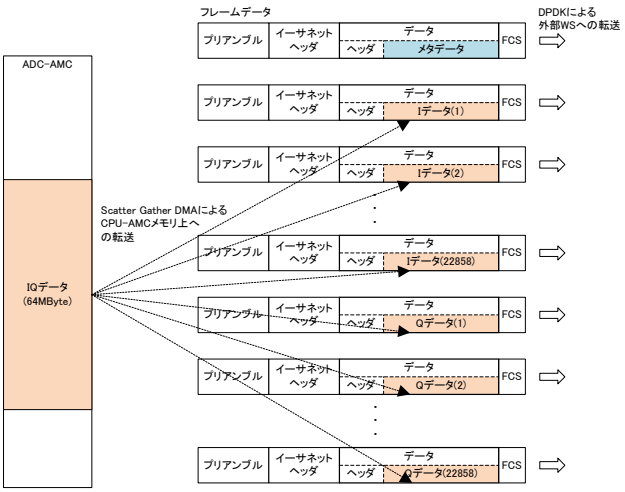

Figure 4: Picture of DPDK of format of data frame.

3.3 DPDK 受信から HDF5 へのリアルタイム保存

MTCA から DPDK で送信された IQ データは WS の NIC に入力され DPDK で受信処理される。DPDK 受信から HDF5 での保存までの処理の流れの概要は次の通りである。

MTCA の 40GbE の NIC から伝送されてきた IQ データを WS で受信する。このデータを HPC で使われている HDF5 のフォーマットにオンラインで変換して、RAM ディスクに保存する。HDF5 は階層化フォーマットであり、データ本体とメタデータで構成される。ここでは、Figure 5 のように、メタデータは ADC カード上の FPGA でリアルタイムに演算して求めた IQ データのスペクトラムのピークから 5 番目までの情報および SNR としており、データ本体は、16[bit]幅の IQ データとした。

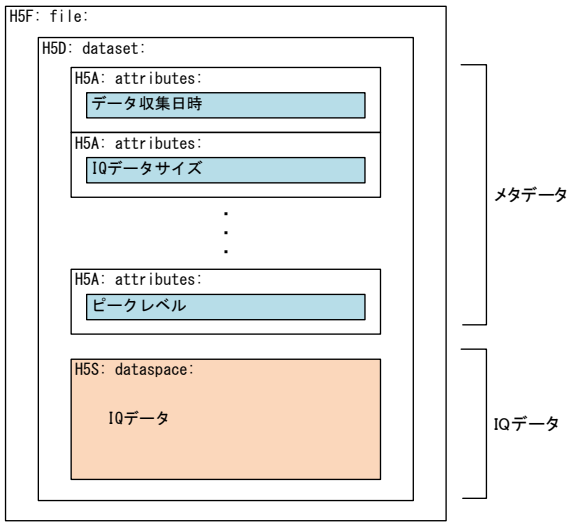

Figure 5: Picture of format of HDF5.

3.4 全系統を連接して伝送速度評価

構築した 40GbE 伝送評価システムの全体で伝送速度の評価をした。ここでは、ADC-AMC 上に収集された 64[MB]の IQ データを 40GbE で伝送し、WS 上で HDF5 へ変換するまでの時間をそれぞれ測定した。Figure 6 に示した測定箇所の結果は次の通りである。

- DMA 伝送速度
  20.7[Gbps]⇒64[MB]*8/20.7[Gbps]≒24.7[msec]
- DPDK 伝送速度
  23.8[Gbps]⇒64[MB]*8/23.8[Gbps]≒22.3[msec]
- HDF5 伝送速度
  17.5[Gbps]⇒64[MB]*8/17.5[Gbps]≒29.3[msec]

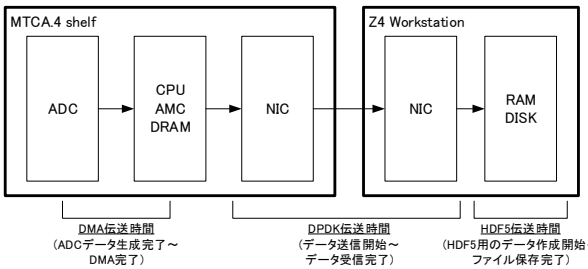

Figure 6: Picture of measurement point of transmission time

DPDK は、以前のサンプルアプリによる評価（23.8[Gbps]）と同等の性能を出せており、ハードウェアの性能を引き出せていると考えられる。しかし、DMA の伝送速度が、以前の評価結果（25.5[Gbps]）の 8 割程度しかでていない。調査の結果、測定方法が異なることが分かった。前回は、FPGA 内の DMA コントローラである XDMA のハードウェアカウンタを使って実際に DMA 転送している期間の伝送効率から速度を算出していた。今回は、ソフトウェアが S/G のリストを作成開始時点から DMA 完了割込みを受けるまでの時間を測定した。

- 前回の DMA 伝送速度
  25.5[Gbps]⇒64[MB]*8/25.5[Gbps]≒20.1[msec]

この 4.6[msec]の時間差は、どのような処理による処理時間なのか調査中である。

3.5 転送速度の改善

現在、DMA・DPDK・HDF5 をシーケンシャルに実施しているため、1 つの IQ データ(64MB)を生成する時間に伝送および保存するための処理時間が間に合っていない。例えば、ADC-AMC にて 500[MSPS]の各 16 [bit]の IQ データ 64[MB]分は 33.7[msec]ごとに生成されるに対し、76.3[msec]ごとにストレージ保存されている。改善策として次のような方法を考えている。

- DPDK 送信用フレームをダブルバッファにしてパイプライン処理化する。
- WS 側の DPDK 受信と HDF5 保存を別スレッドにしてパイプライン処理化する。

S/G DMA を使うことにより、「DMA 実施」=「DPDK 用フレーム生成」となり、処理の短縮ができている。さらに、処理をまとめるため Fig. 7 のように MTCA 側に HDF5 処理を入れると、①「DMA 実施」⇒ ②「HDF5 データ生成」⇒ ③「DPDK 用フレーム生成」という処理になり、今回の構成に比べて②、③の処理がオーバヘッドとなる。これを改善するためには、①「DMA 実施」= ②「HDF5 データ生成」= ③「DPDK 用フレーム生成」とする処理の仕組みの検討が必要となる。実装方法は、FPGA と CPU 処理を最適化して無駄なデータ伝送が生じないように検討を進めていく。

まず、今回の評価には間に合わなかった AMC-CPU をコア数が多く高速処理ができる AMC754 に変更して、ハードウェアによる改善を確認する。

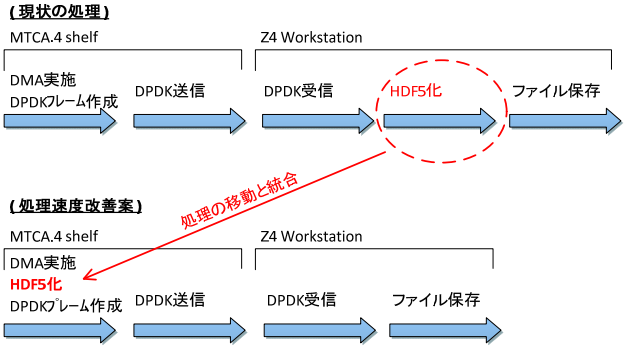

Figure 7: Improvement plan of process.

4. まとめ

加速器制御で使われている MTCA シェルフに ADC-AMC・CPU-AMC・40GbE NIC-AMC を実装し、データ受信用の WS を使って、40GbE の高速・大容量データ伝送の評価をおこなった。データ収集と送信を行う AMC-CPU では、ADC-AMC ボードからの PCIe DMA の伝送先および DPDK の伝送元のバッファを共有し、メモリ上のデータ移動による処理のオーバヘッドを極力なくしてデータ伝送時間短縮を図った。また、データ受信側の WS に DPDK の受信後 HDF5 へ変換する仕組みを構築した。しかし、目標とする伝送速度には達しておらず、改善策としてまず AMC-CPU を当初予定していた高速処理ができるカードを評価してみる。ソフトウェア制御側は、DMA と DPDK の処理をパイプライン化して性能向上を図る。また、複数のデータソースのイベント同期性の確保の仕組みを追加して評価する予定である。

これにより、一連のデータハンドリングシステムが高精細・高分解能のセンサからの高速・大容量のデータを伝送するシステムへの適用が可能になると考えられる。

参考文献

[1] M. Ryoshi *et al.*, “High-Bandwidth data handling system with MTCA.4”, Proceedings of the 14th Annual Meeting of Particle Accelerator Society of Japan, Sapporo, Aug., 2017.