MTCA.4 による高速大容量データハンドリング

HIGH-BANDWIDTH DATA HANDLING SYSTEM WITH MTCA.4

漁師 雅次 #, A), 岩城 孝志 A), 出口 久城 A), 林 和孝 A), 松本 隆太郎 A), 山崎 伸一 A)
Masatsugu Ryoshi #, A), Takashi Iwaki A), Hisakuni Deguchi A), Kazutaka Hayashi A),
Ryuutaro Matsumoto A), Shinichi Yamazaki A)
A) Mitsubishi Electric TOKKI Systems Corporation

Abstract

We have developed various modules corresponding to MTCA and applied it to accelerators control and monitoring. Conventionally, Linux has been installed on the CPU in the FPGA (Virtex 5 FXT, Zynq) implemented in AMC and used as EPICS IOC.MTCA's backplane was used GbE star-topology wiring. Many kinds of LLRF system could be flexibly assembled by the MTCA boards and EPICS. Recently, MTCA applications were expanding using high-speed serial lanes (PCIe or XAUI) on the MTCA backplane. For the monitoring camera device, the data transmission exceeding '100[MB/sec]' was achieved by PCIe Gen2 x4 DMA. We need many input channels and high speed of sensors, so we can supply to use high speed, large capacity data transmission. For that purpose, we confirmed the performance implementation method of PCIe Gen3 and 40GbE systems with MTCA and evaluated the performance.

1. はじめに

高周波加速器の制御システムでは、無線通信における RF 信号処理技術（ディジタル技術）やソフトウェア技術を応用して高度化を進めてきている。その一つとして Advanced Telecommunication Computing Architecture (ATCA)から派生した Micro TCA (MTCA)の利用がある。ATCA の子基板 AMC（Advanced Mezzanine Card）を活線挿抜可能なスロットインモジュールとして構成する MTCA は、運転管理やメインテナンスに適した標準プラットフォームとして採用されている。モジュール間はバックプレーンを介して高速シリアル通信を行っている。port0、1 は GbE、port4～8、9～12 は PCI Express (PCIe)や 10 Gigabit Attachment Unit Interface(XAUI)などの通信規格が適用される。これらの通信は MTCA 専用の HUB である MTCA Carrier Hub (MCH)を中心としたデュアルスタートポロジーでリダンダント構成になっていることが多い。これらの他に port2、3 は SATA・SAS などストレージ用、port13～16 は P2P 通信用で、MCH を介さずにAMCスロット間接続されている。これらに加えて、DESY を中心に広まりつつある RTM の拡張規格である MTCA.4 では、port17～20 を AMC 間の MLVDS 信号によるバス接続で、同期・インターロック信号などに利用されている。

cERL、STF、SuperKEKB では Port0 の GbE を使って EPICS CA により通信をしていた[1]～ [9]。LLRF、BPM、MPCCD や光学実験装置において、Channel 数の増加やセンサの高精細化にともない、扱うディジタルデータ容量も増大する傾向にある。そのため、MTCA シェルフ内のデータ伝送および上位装置へのデータ伝送を高速大容量で行う必要があると考えられる。

そこで、シェルフ内の通信には PCIe Gen3 を、上位装置との通信には 40GbE を使ったシステムの実現方法の確認および性能評価を行った。以降に、その状況を報告する。

ma-ryoshi@west.melos.co.jp

2. MTCA の利用拡大

2.1. SPring-8 における画像処理システム

SPring-8 蓄積リングで使用される 2 次元干渉計の高度化のために、清道らは MTCA を使った画像処理システムを開発した[10]。VGA（640×480）サイズのカメラ画像をカメラリンクで取り込み、1Hz で PCIe の DMA でプライマリプロセッサ AMC へ画像データを送り、1 次元フィッティング処理を行った。その後、セカンダリプロセッサで 2 次元フィッティング処理を行い、この結果を中央制御室に表示している。本システムの実力は 100[fps]であった。

2.2. SPring-8 における DDH プロジェクト

SPring-8 では利用実験の大規模化に伴うセンサ数の増加に対応するために、松下らはバックプレーンの伝送性能およびオープンスタンダードである MTCA を選定してデータ収集機能を組込んだ[11]。

2 つのセンサ出力をカメラリンクで受けて、MTCA のバックプレーン上を PCIe Gen2 x4 で DMA 伝送して 10GbE NIC から上位装置へとデータ伝送した。port4-7 および 8-11 を別の MCH を経由してデュアルスタートポロジーを利用した。この結果、10GbE 一系統で 7.4Gbps のデータ帯域および 40 時間以上の連続動作をした。

2.3. XSBT におけるバンチ形状モニタ

SACLA から SPring-8 へのビーム輸送ラインである XSBT（XFEL to SPring-8 Beam Transport）に設置される 3 次元バンチ形状モニタに清道らは、MTCA のデータ収集装置を利用した[10]。MTCA ダブルハイトの FMC キャリア AMC にカメラリンクの FMC を実装して、6 台のカメラから 60Hz で同時収集する。PCIe Gen2 の DMA を使いプロセッサ AMC へデータを集約し 3D バンチ構造の構築を行う。FMC キャリア AMC には、ARM Cortex-A9 内蔵の Zynq-7000 が実装されており、前処理を行うことでプロセッサ AMC と分散処理をしてリアルタイ

ム再構築を行う。データは 11[MB/frame]で 10[fps]の転送していたため、880[Mbps]の転送レートを実現した。

3. MTCA を使ったさらなる高速大容量データ伝送技術

MTCA を使ったディジタルデータ処理において、シェルフ内の通信はコンピュータ内の通信で一般的になってきた PCIe を使うことが増えてきている。また、外部との通信はコンピュータと接続するときには Ethernet を使うことが考えられるため 40GbE を選定した。それぞれを MTCA シェルフのバックプレーンを活かして利用するための設計手順を確認し評価した。

3.1. 40GbE の評価

MTCA を使ったシステムで 40GbE の通信確認をするために Figure 1 のような評価システムを構築した。ÜBER 製の MTCA.4 対応のシェルフに、PCIe スイッチが拡張された N.A.T.製の MTCA.4 対応の MCH を実装した。この MCH の背面には CPU カードを拡張した。40GbE の NIC には Vadatech 製の AMC を使用した。また対向装置には、HP 製 WS の Z820 を用いて、Intel 製の NIC を実装した。今回使用した NIC に使われている Ethernet Controller は両方とも Intel 製 XL710 である。使用した CPU カードおよび WS のスペックは Table 1 の通りである。

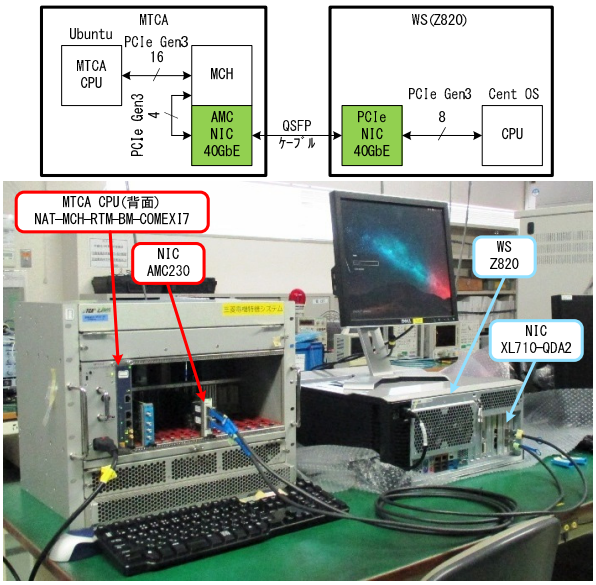

Figure 1: Evaluation Test Environment of 40GbE.

Table 1: Equipment Configuration of 40GbE

	NAT-MCH-RTM-BM-COMEXI7	Z820
CPU	Core-i7 3517UE	Xeon E5-2643
	2Cores,4threads	4Cores,8threads x2CPUs
Memory	4GB	64GB
	ECC DDR3-1066	ECC DDR3-1600
OS	Ubuntu 14.04.4 LTS	CentOS 7.2
	Kernel 4.2.0-27	Kernel 3.10.0-327
NIC	AMC230	XL710-QDA2

最初は、以前設計した 10GbE のソフトウェア構成と同じ BSD Socket を使い UDP 通信で評価を行った。UDP では、10[Gbps]（7.5[Gbps]～12.5[Gbps]）出たが、通信開始から 2～3[分]でタイムアウトにより通信ができなくなり非常に通信が不安定だった。

UDP による通信品質改善のために、UDP 送受信パケットバッファサイズを調整したが、通信速度と品質ともに改善されなかった。

次に TCP では、タイムアウトが起きず通信が安定した。さらに、パケットバッファサイズを調整したものの 10.4[Gbps]より通信速度が早くすることはできなかった。

そこで、ソフトウェアの実装方法を根本的に見直し、Intel が策定した Data Plane Development Kit (DPDK)を使うことにした。ソケット通信と DPDK におけるソフトウェアの構成の比較を Figure 2 に示した。

ソケット通信を使ったソフトウェアの実装方法では、データパケットを送受信すると割り込みが発生する。多くのデータ送受信する場合、大量の割り込みがオーバーヘッドとなり処理が遅くなることが課題となってきた。そこで、一般的になってきたマルチコア CPU の、1 コア以上を通信専用に割り当てることで、NIC の通信用バッファの状態を 100%ロードでポーリングする。また、ソケット通信では kernel driver を使うが、DPDK では userland driver を使用する違いがある。

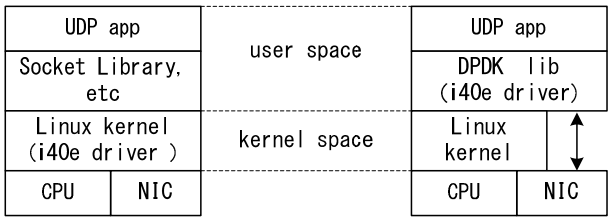

Figure 2: Compare the Configuration of the Socket Communication with DPDK.

DPDK は、H/W 構成に強く影響されるため、DPDK に対応しているかを確認する必要がある。主な確認項目は下の通りである。

- DPDK 対応済み NIC であること。
- マルチコア CPU であること。
- 仮想化対応 CPU であること。
- PCIe revision が伝送レートに十分対応可能なこと。
- kernel version：2.6.34 以降
- kernel configuration：
  CONFIG_HUGETLBFS,
  CONFIG_PROC_PAGE_MONITOR, etc.
- glibc：2.7 以降

DPDK を評価するために、送信パケットを生成する為のソフトである pktgen を使って送信パケットサイズを変化させ、testpmd を使って通信速度の測定をした。

パケットサイズを増やすにつれて、伝送速度がどのように変化をするかを Figure 3 に示した。Z820 から送信して MTCA CPU で受信した場合は、Z820 の送信パケットサイズを 1000[B]まで増やしていくと伝送速度が向上していき、36[Gbps]で速度が飽和した。この時、対向の MTCA の受信速度は、パケットサイズ 250[B]くらいから飽和領域になり 24[Gbps]で頭打ちになった。送受信の伝送速度の差は、送信側に比べて受信側の処理性能が低く、DPDK 自体はプロトコルなしの通信なのでパケットロスしているためだと考えられる。次に、送受信が逆向きの MTCA CPU から Z820 へ送信した場合、MTCA CPU の送信速度は送信パケットサイズ 300[B]くらいから飽和領域になり、24[Gbps]で頭打ちになった。また、Z820 の受信側では、大量のパケットロスが発生して、送信パケットサイズ 300[B]の時に受信データは最大で 10[Gbps] だった。

送信側のそれぞれのデータ伝送速度の差は、送信パケット生成速度の差であり、これはそれぞれの CPU の性能の差が出ていると考えられる。

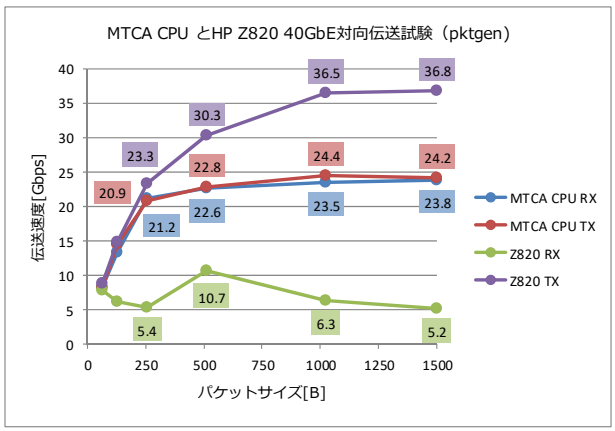

Figure 3: Compare the Transmitting Speed with Packet Size.

Z820 の受信パケットロスが MTCA CPU よりも多すぎる点を調査した。まず、Z820 で pktgen を起動するときに割り込み関連のエラーが発生していた。その為、dmesg の内容通り割り込みがでないように kernel parameter に irqpoll を追加したが、改善されなかった。次に、Z820 の Linux のディストリビューションが MTCA CPU のものと異なったので、同じ Ubuntu に合せたが改善はみられなかった。そのため、ハードウェアによる違いと考えて調査を進めた。

Z820 はデュアル CPU で構成されており、CPU ごとにシステム・メモリが接続されており、それぞれがノードとよばれ区分けされている。すべてのシステム・メモリは、いずれの CPU からでもアクセス可能とするために CPU 間がインターコネクトで接続されている。同一ノードへのアクセスよりも、異なるノードへのアクセスには時間がかかる、非対称な構成になっている。このような構成を Non-Uniform Memory Access (NUMA)と呼ぶ。

Z820 では、ノード 0・1 の 2 つのノードがあり、ノードの下に 4 つの物理コアがある。DPDK を使った設計当初は、NUMA 構成で OS、DPDK 送信・受信の機能を、OS で 1 コア、送信・受信でそれぞれ 2 コアずつ割り当てていた。そのため、ノードを超えた物理コアの割り当てになっていた。Z820 の受信パケットロスが多い原因をノードをまたいだコアの割り当てだろうと推測して、送信・受信でそれぞれ 1 コアずつ割り当てに変更した。結果、Figure 4 のようになった。NUMA ノード 0 側のコアだけを使うようにしたところ 24[Gbps]で受信できるようになり、パケットロスがほぼなくなったことを確認した。

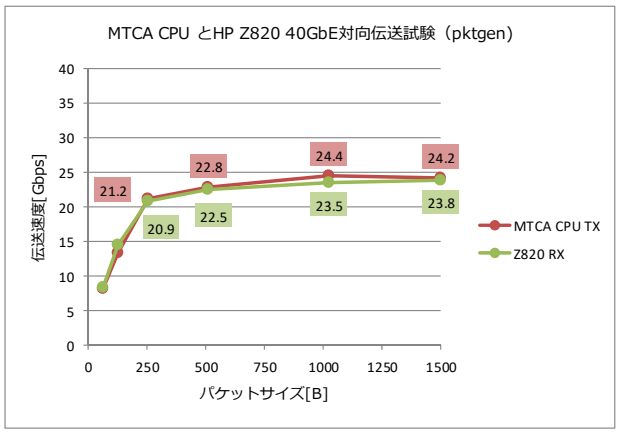

Figure 4: Compare the Transmitting Speed with Packet Size after the NUMA Configuration Changed.

現在のハード構成での限界を評価するために、NIC にある 2 ポートを、両方とも接続したときの伝送速度を確認した。Z820 の送信の伝送速度は 45[Gbps]まで向上したが、MTCA 側の伝送速度は 1 本の時と同様で 24[Gbps]だった。このことから、MTCA CPU カードの性能限界だと考えられる。

3.2. PCIe Gen3DMA の評価

コンピュータ内の CPU と周辺機器の間でデータ通信するためには高速シリアル通信の PCIe が一般的に利用されている。最初は片方向 2.5[Gbps/lane]の Gen.1 だった。次に片方向 5[Gbps/lane]の Gen.2、そして最新である 8[Gbps/lane]の Gen.3 と発展してきた。最近では、外

部のコンピュータとケーブルを使って接続され通信するような使い方もされるようになってきた。

PCIe Gen3 インタフェイスを備えた AMC を開発する場合、PCIe Gen3 専用ブリッジ ASIC を使う方法もあるが、AMC 上で様々なディジタル信号処理をすることを考えると、部品点数が削減できフレキシブルな論理を組込むことができる FPGA を使用した設計がよいと考えた。

そこで、従来から使っている Xilinx 製の FPGA をターゲットに考えた。AMC を EPICS IOC にするために内蔵 CPU 上で Linux を使うことを考え Zynq を使っていた。しかし、Zynq は PCIe Gen2 までしか対応しておらず、PCIe Gen3 に対応するには UltraScale Kintex 以上が必要となる。そこで、Figure 5 に示したとおり Xilinx 製の評価ボード KCU105 を Z820 に実装して性能評価をした。

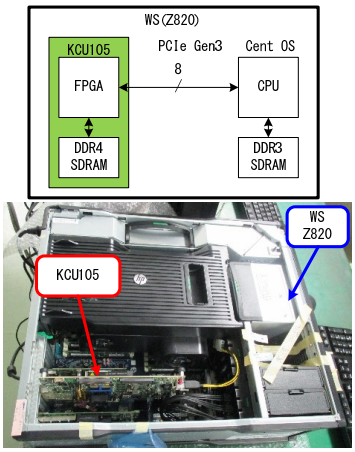

Figure 5: PCIe Evaluation Test Environment of KCU105.

まず、KCU105 の PCIe Gen3 のリファレンスデザイン資料「ug919」を参考にして評価回路を準備した。このリファレンスデザインは、「Northwest Logic 製の DAMC-IP」を使ったデザインである。これにより、FPGA と評価環境の性能の基準を確認した。

次に、Xilinx 製の標準 DMAC-IP を使った評価回路を設計した。特に設計上は NL 製の IP と違いはなかった。両者とも、KCU105 上の DDR4-SDRAM と Z820 上の DDR3-SDRAM 間の PCIe Gen3 の DMA 伝送を行い、データ伝送速度の評価をした。伝送速度は、FPGA 内蔵の PCIe のハードマクロの測定機能を使った。

MTCA では通常 4lanes の PCIe が使われる。今回評価に使用した KCU105 は 8lanes の PCIe Gen3 で構成できる。lane 数はハードウェアおよびドライバの設計にあまり大きな影響を与えないため 8lanes で評価した。

NL 製の IP のリファレンスデザインで評価用のソフトを用いて送受双方向通信で実測した。送信および受信それぞれで約 47[Gbps]だった。これは、Z820 に使用しているメモリが DDR3-1866 であり、理論転送速度 119.2[Gbps]であり、伝送効率 78.9[%]とリードライトアクセスを繰り返していると考えるとほぼ上限に達しているため、メモリバスで帯域制限されていると考えた。よって、今回の評価構成において PCIe の伝送速度のみを評価する場合、双方向同時通信ではなく、片方向通信のみで評価した方がよいと考えた。

そこで、片方向通信のみで評価したところ、平均して約 54[Gbps]とデータ伝送速度が向上した。

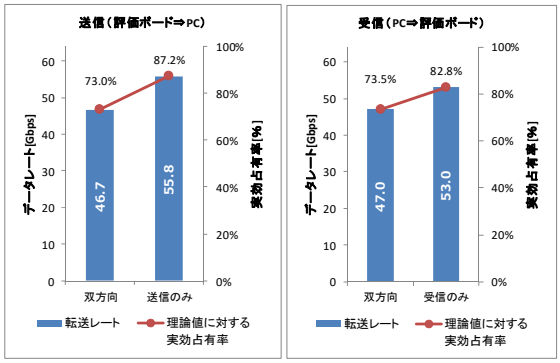

Figure 6: Compare the Transmitting Speed of Bi-directional to the Transmitting Speed One Way by NL IP

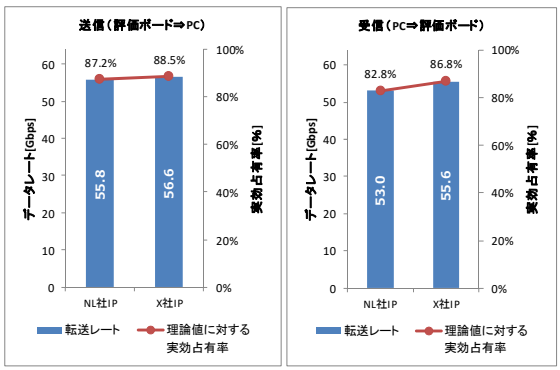

Figure 7: Compare the Transmitting Speed of Xilinx DMAC-IP to NL DMAC-IP.

Xilinx の DMAC を使った回路で片側ずつ動作させて評価したところ平均して約 56[Gbps]とさらに向上した。違いは、NL 製の DMAC に比べて Xilinx 製の DMAC のリリース時期が新しいくらいしか分かっていない。

MTCA バックプレーンを使って PCIe Gen3 を評価するために、KCU105 に実装されている FPGA と同じ型名の UltraScale Kintex を使った高速 A/D 変換の AMC を開発した。本カードは、分解能 12[bit]、4[GSPS]の A/D 変換性能があり、バックプレーンとは、Port4-7 を使って PCIe Gen3 4lanes の接続がある。本 AMC-ADC を MTCA シェルフに実装して DMA 伝送性能を測定した。

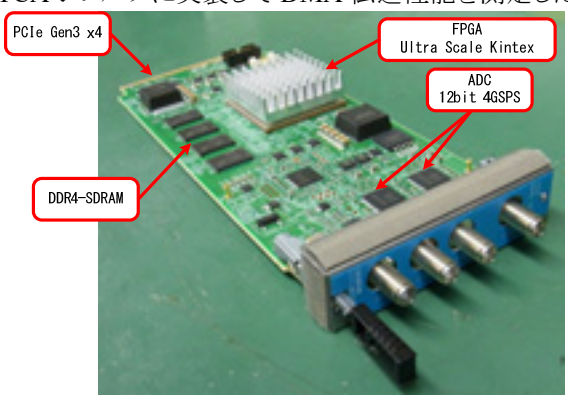

Figure 8: Picture of 12bit-4GSPS A/D AMC with PCIe Gen3 4lanes.

DMA 転送した結果を Figure 9 に示す。4lanes の AMC-ADC による伝送速度と比較するために、Z820 と KCU105 で Xilinx 製の DMAC-IP を使い 8lanes で評価した伝送速度を半分にした。その結果、AMC-ADC の伝送速度は、KCU105 の 9 割程度に相当する 25.5[Gbps]になった。両者とも FPGA 側のメモリは DDR4-SDRAM の 64[bit]接続であり十分な帯域がある。よって、40GbE の評価結果と同程度の伝送速度限界であったため MTCA CPU カードの性能限界と考えられる。

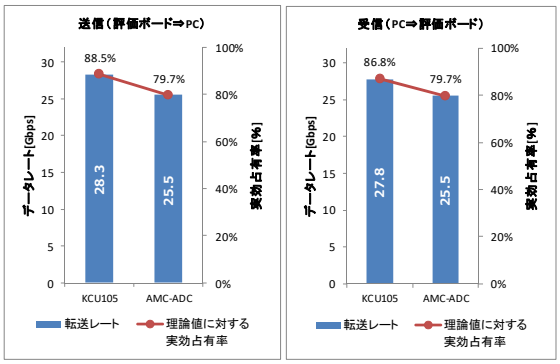

Figure 9: Result of the Transmitting Speed of AMC-ADC.

3.3. 今後の予定

PCIe DMA と 40GbE DPDK の組み合わせた伝送の評価を準備している。例えば、AMC-ADC で収集したデータを PCIe DMA してそのまま DPDK を使った 40GbE にて上位コンピュータへデータ伝送する。簡単には、DMA の転送先バッファと DPDK 送信バッファを別に確保して、メモリコピーをしてつなげることが考えられる。しかし、大容量のデータ伝送を考えた時に、このコピー操作はレイテンシも大きくなるし、リソースの無駄遣いや連続的な伝送ができなくなる可能性がある。そこで、二つのバッファを共有することを考えてドライバを構成する。まずは、少量のデータ容量の伝送を試行して、バッファの共有ができた。

しかし、ここでも技術的なハードルが出てきた。DPDK は VFIO 使って IOMMU(IO Memory Management Unit)を利用するので user 空間から制御する。しかし、DMA ドライバは kernel 空間で動作するため、同じ IOMMU group では動作させられない。MTCA シェルフは 1 つの IOMMU group で構成されているため、DPDK を noIOMMU モードで動作させるように検討している。

4. まとめ

MTCA シェルフを使った構成で、PCIe Gen3 の DMA で 25Gbps 以上、40GbE で 23Gbps 以上のデータ通信ができることを確認した。高速大容量データの取り扱いが可能となり、LLRF やモニタの新たな可能性と用途を広げていく。40GbE ではパケット送受信の割り込みによる影響が大きいためポーリングを利用した DPDK を使って評価した。PCIe では、UltraScaleKintex を使って DMA 転送技術を評価した。今後、FPGA を使った PCIe の DMA と 40GbE を組み合わせることで、FPGA で処理したデータを PCIe で CPU 上のメモリに DMA して、そのデータを 40GbE で外部のコンピュータへ伝送する技術開発および評価を進めていく。そして、複数の AMC にわたって収集したデータを統合して判断して、制御に生かすことへと用途を広げていけると考えている。また、画像データのような大容量データを取り扱うシステムへ応用するなど適用範囲の拡大を図ってきたい。

参考文献

[1] M. Ryoshi *et al.*, “LLRF Board in Micro-TCA Platform”, Proceedings of the 7th Annual Meeting of Particle Accelerator Society of Japan, Himeji, Aug., 2010.
[2] T. Miura *et al.*, “Digital feedback system using μTCA for DRFS", Proceedings of the 8th Annual Meeting of Particle Accelerator Society of Japan, Tsukuba, Aug. 1-3, 2011.
[3] M. Omet *et al.*, “Development and Application of a Frequency Scan-based and a Beam-based Calibration Method for the LLRF Systems at KEK STF”, Proceedings of the 9th Annual Meeting of Particle Accelerator Society of Japan, Osaka, Aug. 8-11, 2012.
[4] T. Kobayashi *et al.*, “Prototype Performance of Digital LLRF Control System for SuperKEKB”, Proceedings of the 8th Annual Meeting of Particle Accelerator Society of Japan, Tsukuba, Aug. 1-3, 2011.
[5] S. Michizono *et al.*, “Tuner control for cERL cavities by digital feedback system", Proceedings of the 9th Annual Meeting of Particle Accelerator Society of Japan Osaka, Aug. 8-11, 2012.
[6] H. Ishii *et al.*, “Development of a beam position detector for an orbit feedback system in SuperKEKB”, Proceedings of the 8th Annual Meeting of Particle Accelerator Society of Japan, Tsukuba, Aug. 1-3, 2011.
[8] T. Kobayashi *et al.*, “RF Reference Distribution System for SuperKEKB”, Proceedings of the 10th Annual Meeting of Particle Accelerator Society of Japan Nagoya, Aug. 3-5, 2013.
[9] K. Hayashi *et al.*, “Refinements of the new LLRF Control System for SuperKEKB”, Proceedings of the 9th Annual Meeting of Particle Accelerator Society of Japan Osaka, Aug. 8-11, 2012.
[10] A. Kiyomichi *et al.*, “Improvement of MICROTCA-Based image Processing System at SPRING-8”, Proceedings of the 12th Annual Meeting of Particle Accelerator Society of Japan, Tsuruga, Aug. 5-7, 2015.
[11] T. Matsushita *et al.*, “SPring-8 Annual Report 2014”, 2014.