

IMPROVEMENT OF DATA TRANSFER SPEED OF LARGE MEMORY MONITORS

M. Tobiyama*, KEK Accelerator Laboratory, 1-1 Oho, Tsukuba 305-0801, Japan, and
Graduate University for Advanced Studies (SOKENDAI), 1-1 Oho, Tsukuba 305-0801, Japan

Abstract

Beam monitors with long memories will be widely used in SuperKEKB accelerators. Since the slow data transfer time of such devices usually limits the operational performance, improvement of the transfer rate is required. Two kind of devices, VME-based modules and Ethernet-based modules have been developed. On the VME-based devices such as turn-by-turn position monitors for damping ring or long bunch oscillation monitors, MBLT and BLT transfer method have been implemented. For the Ethernet based system, such as the gated turn-by-turn monitors (TbT), SiTCP has been implemented on the FPGA and the EPICS device support for SiTCP has been developed. The improvement of the data transfer speed with the long-term reliability will be presented.

INTRODUCTION

With the rapid development of digital technology, it is much easier to implement large memory in the beam monitor system since late 1990's where we had designed the beam monitor devices for KEKB accelerators. As new accelerators with strongly improved performance such as SuperKEKB need much detailed information of the beam to achieve the design characteristics, the number and scale of the beam monitors with long memory becomes much larger than that of KEKB accelerators.

On the other hand, the environment of the beam instrumentation such as the selection of the field bus tends to be fairly conservative mainly due to saving of the design and construction efforts and costs. For example, in SuperKEKB beam instrumentations, though we newly introduce MTCAs and modules with direct Ethernet connection in some limited sections, we will still use the legacy bus such as VMEbus, VXIbus and GP-IB.

In the operational view, it is needless to emphasize the importance of the fast data acquisition and the fast data processing to minimize loss time of the operation. For example, during the operation of the KEKB collider, we have used the VMEbus based bunch oscillation recorders (BOR) with total 20 MB of memory which enables us to record the 4k turns of bunch oscillations for all the buckets in the ring [1]. We have also used the same recorders with limited address space of 5120 bytes for the bunch current monitors (BCMs). The BORs have been used for the post-mortem analysis of the beam abort and the machine developments such as the detailed analysis of the beam oscillation coming from intra-bunch oscillation due to electron cloud instabilities [2]. BCMs have been

used to support the injection bucket selection systems to realize the equally filled bunch filling. As the EPICS system we had mainly used during the KEKB operation (R313) did not support larger array than 4k bytes, we needed to store the data to the remote disk directly from the IOC through the fairly slow Ethernet line (10base connection). Typical data transfer time from the BOR to the disk was 5 min to 10 min depending on the network traffic and the CPU usage of the host workstation with the remote disks. Of course that kind of waiting time was painful for all of us and had spoiled the efficiency of the machine operation.

For the SuperKEKB accelerators, we have developed the beam instruments with improved data transfer speed. Two kind of devices, the VMEbus based modules and direct Ethernet connection modules will be shown here. Table 1 shows the main parameters of SuperKEKB accelerators, main rings (LER and HER) and the positron damping ring (DR).

Table 1: Main Parameters of SuperKEKB Rings

	HER/LER	DR
Energy (GeV)	7/4	1.1
Circumference(m)	3016	135.5
Beam current (A)	2.6/3.6	0.07
Number of bunches	2500	4
Single bunch current (mA)	1.04/1.44	18
Bunch separation (ns)	4	>98
Bunch length (mm)	5/6	6
RF frequency (MHz)	508.887	
Harmonic number (h)	5120	230
T. rad. damping time (ms)	58/43	11
L. rad. damping time (ms)	29/22	5.4
Number of BPMs	466/444	83
Number of TbT monitors	135/135	83

VMEBUS BASED SYSTEM

Though the specifications and the expected performance of the VMEbus system might not be so modern [3], still the system have many strong points as:

- Board size fits most of our purpose as the beam instrumentation devices.

*makoto.tobiyama@kek.jp

- Fairly simple bus interface and bus protocol. Bus interface could be realized with a simple CPLD-based logic ICs.
- There exist enough accumulation of usable resources such as already developed boards and operational experiences. It is still possible to obtain commercial boards with reasonable price.

In SuperKEKB, though the CPU (IOC) will be replaced from old PPC6750 to MVME5500, the VMEbus systems will still be used as the main field bus. The EPICS system will also be upgraded from R313 to R314 with VxWorks 6.8.2 which intrinsically supports large waveforms more than 20 MB. Therefore, it is a big concern about the total data transfer rate and the rate-limiting part in the data acquisition system.

Beam Position Detector for Damping Ring

In the normal operation mode of the DR, we inject one or two positron bunches of 1.1 GeV from the linac, damp the emittance during at least 40 ms of revolution, then eject them to the latter half of the linac to accelerate the beam up to 4 GeV. As the maximum repetition rate of the positron injection to the LER will be 50 Hz, 2nd set of positron bunches (single or two) will be injected 20 ms after the previous injection. It is therefore not suitable to use a narrowband position detector which needs longer accumulation time. We have developed the turn-by-turn beam position detector based on log-ratio method on the single VMEbus board (18K11) with the memory size of 32k to 256k words per channel which corresponds the recording time of 14.5 ms to 118 ms [4]. As there will be four BPM handling stations in the ring, one station needs to handle 20 to 22 BPMs. We will prepare two VME64x (without P0 connector) subracks for each station: one VME subrack contains 10 to 11 18K11s. Figure 1 shows the 18K11s in a VME64x subrack with a MVME5500 CPU as an IOC.

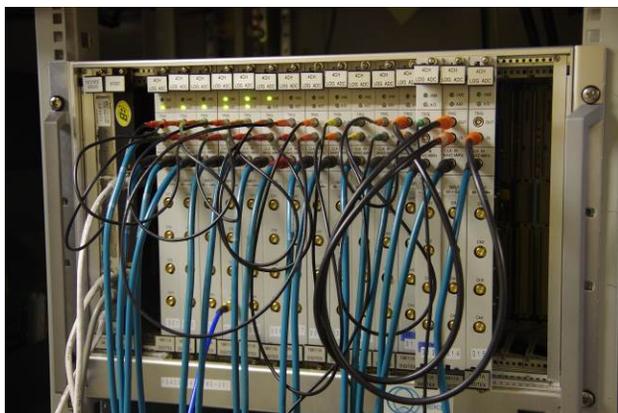


Figure 1: Log-ratio beam position monitor system (18K11) at the calibration station.

The data transfer speed of MVME5500 CPU with simple I/O mode (AM=0x0d) has been measured using a VME bus analyzer (HP 16500C or Agilent 16803A + FuturePlus FS3100) as shown in Fig. 2. The normal data

accessing cycle is around 2 μ s, that means it needs more than 0.15 s to transfer 4ch of 32k turns of data to the IOC. If we install 10 of 18K11s in one VME subrack, maximum repetition rate of the measurement will be smaller than 0.5 Hz.

To speed up the data transfer from 18K11 to the VME IOC, we have implemented A32 supervisory 64-bit block transfer (MBLT, AM=0x0C) defined by VME64 specification to 18K11 and developed the control code for UNIVERSEII PCI to a VME bridge chip on MVME5500 CPU. In the code, we at first transfer the specified memory block of the VMEbus module to the SDRAM of the CPU local memory using MBLT(VME) and DMA(CPU), then transfer them to EPICS waveform data.

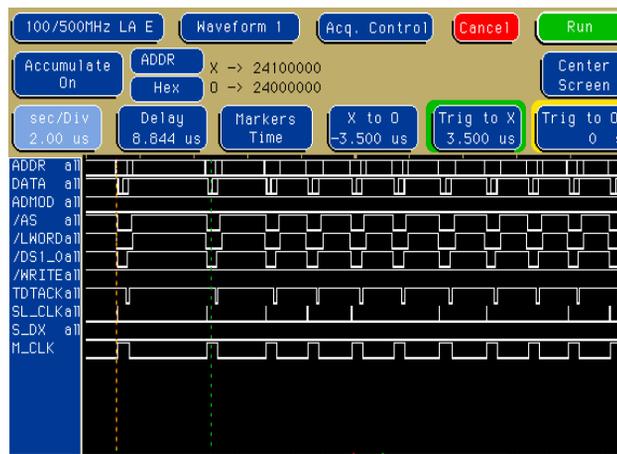


Figure 2: A32 supervisory data access using MVME5500. Typical bus access interval was around 2 μ s.

By using the MBLT access, the bus access interval has shrunk down to 500 ns as shown in Fig. 3. During the MBLT cycle, CPU at first places the start address and the AM code of 0x0C (A32 MBLT) on the bus. After the return of DTACK signal of the slave board (18K11), MBLT cycle starts by handshake using DS0/1 and DTACK, and will continue up to the specified number of cycles (up to 256 cycles in one time).

In the real operation, 18K11 starts the data acquisition with the external hardware trigger signal, then initiates the VMEbus interrupt (IRQ) after filling the specified memory length to require data transfer to the IOC. Typical response time of IRQ to the start of the data transfer was about 24 μ s. Data transfer of 32k turns of 4-ch data from 18K11 to the IOC took about 17 ms. After the raw data transfer, it converts the raw channel-mixed data array to EPICS waveform with the form of X, Y positions which needs about 7 μ s. In case of 12-boards for one VME subrack, it needs roughly 0.3 s to transfer all 32k data. We will have enough margin for the 1 Hz system cycle even in the case of 64k turns (= 29 ms of beam position) of data acquisition.

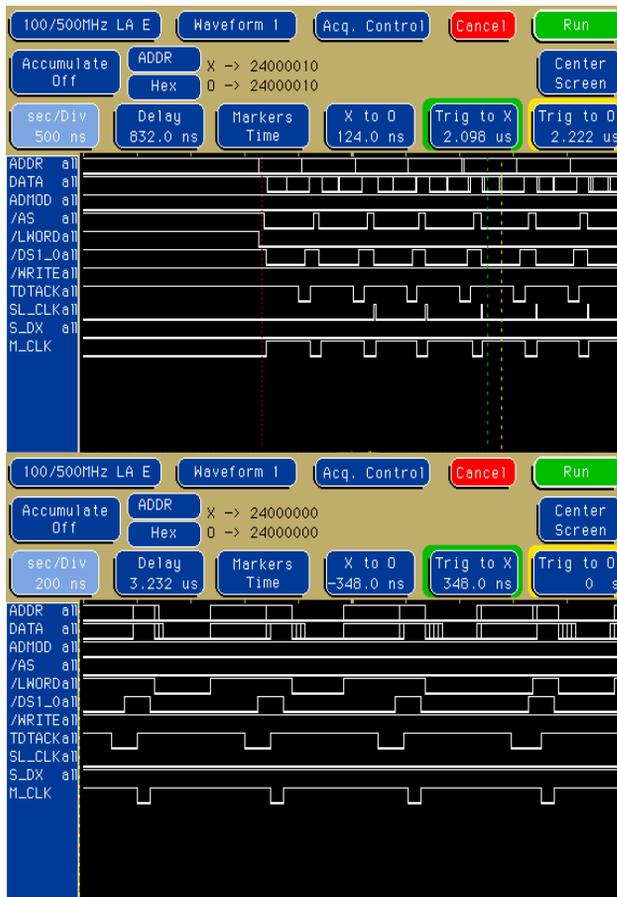


Figure 3: A32 MBLT access. Typical data bus access interval has shrunk to 500 ns (upper). The bus control has been controlled by the DTACK and DS0/1 during the MBLT cycle.

Bunch Current Monitor and Bunch Oscillation Recorder

In colliders such as SuperKEKB, the equalization of the bunch current is very important not only to suppress the instabilities, but also to maintain high luminosity with acceptable beam lifetimes. To control the bunch current during the injection process, it is necessary to measure the bunch currents and decide which bucket should be injected by the next pulse within the injection period, which is 20 ms at the maximum injection rate [5].

The transient behaviour of the beam just after closing or opening of the bunch feedback loop reveals many important characteristics of the coupled-bunch motions as well as the performance of the feedback systems (transient-domain analysis of the instability) [6]. Also the bunch motions just before the beam abort shows the cause of beam abort such as the growth of the instability or sudden beam kick by the miss-fire of the pulsed magnets (post-mortem analysis of the beam abort).

In KEKB, we have prepared 8-bit large scale memory boards with the memory size of 20 MB (the bunch oscillation recorder: BOR) for each planes (X, Y, and longitudinal) and the bunch current monitor board (BCM)

that have same hardware structure as BOR but limited memory address space down to 5120 bytes.

In case of BCM of KEKB, the data transfer speed of 5120 bunches to IOC and reflective memory was about 1.4 ms under PPC6750 IOC with EPICS R313. As the minimum injection interval was 20 ms, there existed enough time to process the bucket selection before the next injection.

On the other hand, the data transfer of 20 MB of data from BOR to IOC took about 16 s with D16 data transfer mode. Because of fairly slow network (10base) coming from PPC6750, the data transfer from IOC to the remote disk (tftp protocol) on the host CPU took about 5 to 10 min depending on the total traffic on the network. As we normally took X and Y positions after the beam abort, it needed at least 10 min before the restart of the BOR systems. The situation became much worse when we took many BOR data in a short period such as machine study of electron cloud [2]; it needed huge time to store the data due to terribly increased network traffic between the IOC and the host CPUs.

We have developed new BCM/BOR system (18K10) based on fast FPGA technology [7]. It mainly consists of a fast 8-bit ADC (MAX108), a Spartan6 (XC6SLX45) daughter card with the form of a SO-DIMM card (Mars MX1), and VME-IF CPLDs and ICs. They are mounted on a double width, 6U VME card. For the BCM mode, as it is not needed to have large memory space, we have implemented the FPGA code to use only the block memory on the Spartan6 FPGA. For the BOR mode, as the MX1 board has 128MB of a DDR2 SDRAM, we have implemented the FPGA code to use the external DDR2 memory. The memory size for SuperKEKB mode ($h=5120$) could be 4k turns (20 MB) to 16k turns (80 MB) with the current configuration. Figure 4 shows the photo of the 18K10 BCM/BOR. As we designed 18K10 as the simple VME module which completely separates the address lines and data lines, it was difficult to implement MBLT on the board. We therefore implemented only simple A32 block transfer mode ($AM=0x0f$). The IRQ mean response time was 8.9 μ s and the data transfer started after 36.5 μ s of the IRQ. In the BCM mode, transferring 5120 bytes to IOC took only 1.1 ms.

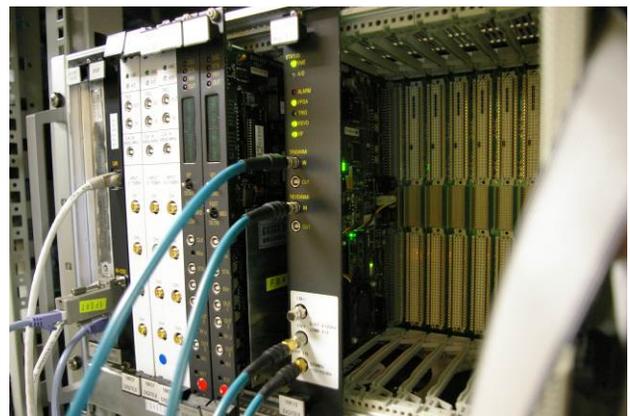


Figure 4: 18K10 bunch current / bunch oscillation monitor.

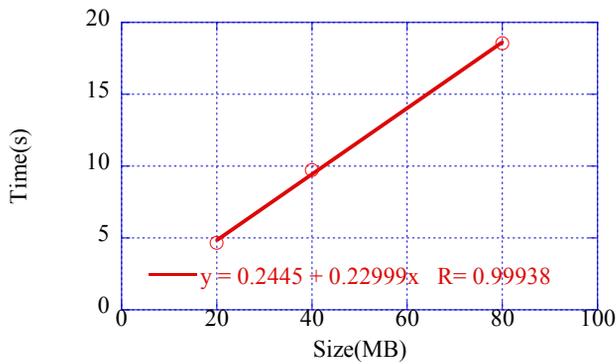


Figure 5: Measured data transfer time from 18K10 to IOC.

For the large data transfer on BOR mode, we have measured the transfer time of 20, 40, 80 MB of data using the VMEbus analyzer as shown in Fig. 5. It roughly shows 4 Mbytes/s data transfer rate.

We have also measured the data transfer speed from the IOC to the remote network disk (NAS, Thecas N8900 with 10GbE IF) mounted using NFS. Though the network speed from MVME5500 IOC to the local edge switch was limited by GbE, the other part, from the edge switch to the NAS, was connected with 10GbE so that we have expected full GbE speed. Figure 6 shows the measured data transferring time measured by the VME bus analyzer.

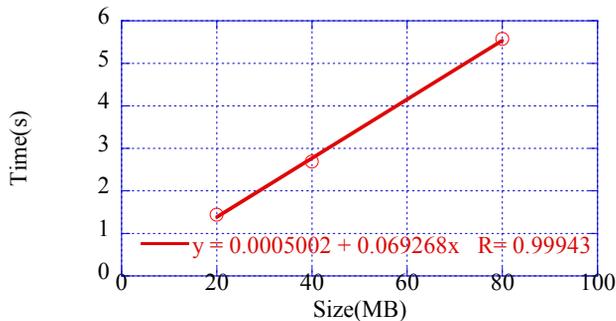


Figure 6: Measured data transfer time from 18K10 to NAS.

It needed less than 6 sec even in the case of 80 MB of data transfer. Though the total time needed to store the large 18K10s data is much shorter than that of old BOR and, waiting time, say 7 sec per one plane (20 MB) might be acceptable, it is obvious that data transfer in the VME system is the strong limiting factor of the total data transfer.

ETHERNET BASED SYSTEM

We have developed the gated turn-by-turn monitor (1421B) mainly to measure the beam optics (phase advance between the BPMs) during the collision using non-colliding bunch without feedback, and to measure the injection beam orbit [8]. It has four independent channels, each of them consists of a fast RF switch, a log-ratio beam signal detector, and a 14-bit ADC. Timing generation, data acquisition and data transfer is controlled

by a Spartan6 FPGA (XC6SLX100T). Raw and calculated position data up to 1M turns per channel is stored in the DDR3 SDRAM. The block diagram and the photo of 1421B are shown in Fig. 7 and 8. We plan to install 12 to 15 1421Bs in one local control room under one server controlled by EPICS (R314.12.3) on CentOS 6.5-64bit system.

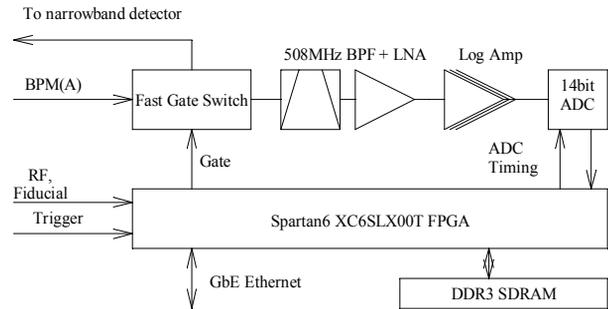


Figure 7: Block diagram of the gated turn-by-turn detector (1421B).



Figure 8: Photo of the 1421B gated turn-by-turn monitor.

Giga-bit Ethernet connection is used to control the system. We at first implemented a MicroBlaze soft processor on the Spartan6 FPGA to handle the commands from the server, and to transfer the data to the server through Ethernet. The measured data transfer rate has been found unexpectedly slow on the prototype system even in the case of 1:1 direct connection through a GbE non-intelligent SW-hub. To transfer 0.5 M turns of data of 4-channels which corresponds 5 sec of turn-by-turn data, it took about 44 s. Even by taking into account the data structure, the data transfer rate was only 2Mbit/s. For injection tuning, typically less than 32 turns of data transfer, or short data around 4k, that slow transfer might be acceptable. Nevertheless it is surely unacceptable to wait longer transfer time during operation. As it is suspected that the MicroBlaze is the main time-limiting principal, we have decided to implement SiTCP [9].

SiTCP supports hardware TCP/IP Ethernet communication from 10Mbps to 1Gpbs Ethernet with features:

- High speed communication stable at the upper limit of TCP.

- Slow control function using UDP.
- Small circuit scale.
- Provided as FPGA library (Xilinx only: no source code available).

The SiTCP was originally developed in KEK and has been technology-transferred to Bee Beans Technology Co. Ltd. (BBT) [10]. The cost per module (mostly registration fee for MAC address) is not expensive. We have also implemented the firmware uploading function through Ethernet which enables us much faster rewriting of a SPI-flash than using JTAG connection. We have kept MicroBlaze to handle or monitor slow data handling in the FPGA, which is not controlling Ethernet communication, such as UART communication, which helps to set the IP address at the very beginning of the operation or after miss-setting the IP address. The booting time of the 1421B is less than 2 s, fairly fast enough. The communication and automatic negotiation with 1000baase and 100base has been confirmed to be OK.

On SiTCP, it is needed to use UDP as the slow control, such as setting some parameters or reading simple parameters, and TCP for larger data transfer. As existing framework provided by EPICS, such as ASYN, does not fit for the device support of SiTCP device, we have developed a mixed socket communication code as the EPICS device support to communicate with 1421B.

During the initialization of the device support, it generates communication threads (t1421ComXX) with the same number of 1421B to connect. In the normal command such as writing or reading short data, the record support at first posts the communication request to the t1421ComXX thread. The communication thread sends command to 1421B through UDP and receives the response also through UDP. It sends the response data to the record support through a callback thread.

In the case of large data transfer, the waveform record support also at first posts the communication request to the t1421CommXX thread. This time the thread tries to open the TCP socket and checks the DMA ready status of 1421B (if failed, it closes the socket and tries again), sends the DMA request through UDP, and receives the data through TCP. After receiving all the data, it closes the TCP socket and sends the data to the record support through a callback thread.

The data transfer rate has been measured using a software timer on the host PC. It was about 400MBPS from 1421B to the host, and about 280MBPS from the host to 1421B under GbE connection. As this asymmetry comes from the intrinsic structure of SiTCP, enough data transfer rate has been realized.

We have also examined long-term, fairly higher load (frequent large data transfer) test on 1421B and the server. During the test, we have encountered several unexpected communication errors and hung-ups of 1421B mainly during the TCP data transfer. As SiTCP is black box for us, we have implemented error handling and recovery functions in the communication thread. Though the error rate has been greatly reduced after this, it still remains communication hung-ups with very rare rate, say, once a

week. We are discussing with BBT and also planning to implement a remote reboot function on the 1421B firmware.

SUMMARY

We have examined the fast data transfer on VMEbus and direct Ethernet devices. For the VMEbus devices, the 18K11 log-ratio beam position detector for DR and the 18K10 bunch current and bunch oscillation recorder, MBLT and BLT defined VME64 has been applied and more than 8 times of faster data transfer on MBLT case has been shown. Also the data transfer from IOC to the remote disk has been improved by the faster network (10GbE) and fast NAS and has shown about 300 times faster rate than that of KEKB.

For the direct Ethernet connection devices (1421B gated turn-by-turn monitor), we have implemented SiTCP to remove huge overheads in the FPGA code. The data transfer speed has been increased about 100 times than before. Also the EPICS device support routine to handle mixed UDP and TCP communication on SiTCP has been developed and has shown excellent performance.

The DMA data transfer code on MVME5500 has been developed by Mr. T. Okazaki of East Japan Institute of Technology Co., Ltd. (e-JAPAN IT Co., Ltd.). The EPICS device support for SiTCP has been written by Mr. Y. Iituka of e-JAPAN IT Co., Ltd. Implementation of SiTCP on the 1421B FPGA has been done by Skywave Co. Ltd. The author would like to thank Dr. T. Obina and Dr. H. Uchida for the discussion of SiTCP implementation. The author thanks colleagues of the SuperKEKB beam instrumentation group for numerous supports on the development.

REFERENCES

- [1] M. Tobiyama, E. Kikutani, Phys. Rev. ST Accl. Beams 3, 012801 (2000).
- [2] J. W. Flanagan, et. al., Phys. Rev. Lett. 94, 054801 (2005).
- [3] W. D. Peterson, "The VMEbus Handbook", ISBN 1-885731-08-6.
- [4] H. Ikeda, et. al., in proceedings of IBIC2013, Oxford, UK. p.53 (2013).
- [5] E. Kikutani, et. al., in proceedings of ICALEPS99, Trieste, Italy (1999).
- [6] M. Tobiyama, et. al., Phys. Rev. ST Accl. Beams 9, 012801 (2006).
- [7] M. Tobiyama and J. W. Flanagan, in proceedings of IBIC2012, Tsukuba. (2012)
- [8] M. Tobiyama, et. al., in proceedings of IBIC2013, Oxford, UK. p295 (2013).
- [9] Tomohisa Uchida, IEEE Trans. Nucl. Sci. Vol 55, No.3, p.1631, (2008).
- [10] <http://www.bbtech.co.jp/>