THE DIAGNOSIS OF EVENT TIMING SYSTEM IN SuperKEKB LINAC*

D. Wang[1,2†], M. Satoh[2], K. Furukawa[2], T. Kudou[3], S. Kusano[3], Y. Iitsuka[4]
[1]SOKENDAI, Kanagawa 240-0193, Japan
[2]KEK, Ibaraki 305-0801, Japan
[3]Mitsubishi Electric System & Service Co., Ltd, Ibaraki, Japan
[4]East Japan Institute of Technology Co., Ltd, Ibaraki, Japan

Abstract

We introduced MRF event timing system in injector linac in SuperKEKB to satisfy our demands. The event timing system is utilized to distribute high-level precise timing signals and accompanying control instructions to synchronize different subsystems and machines. EVG generates beam pulse pattern 50 Hz which contains several event codes while EVR receives them. The Event rate is 114.24 MHz thus the minimal event time interval is about 9 ns. To certain that events are consistent between EVG and EVR, recording them one by one is essentially needed. Owing to some hardware and network restrictions it is difficult to continuously send every event by EVR through EPICS Channel Access. An EVR based events diagnostic system is thus developed by modifying the device support of some records as well as EVR driver mrfioc2 to send the event codes thus comparing the received event codes with the beam-pattern control orders from beam operation and detecting the event timing interval fault as well as providing a logging system of persistent event data. Then, we are able to locate the fault, analyze the data, fix bugs or replace hardware and resume accelerator operation quickly.

INTRODUCTION

The SuperKEKB project is an electron/positron double ring collider upgraded from KEKB project since 2010, endeavoring to update the world highest luminosity record and discover new particle physics by Belle II experiment. The beam commissioning which divided as three Phases started from 2016. During Phase-2 commissioning in 2018, a new 1.1 GeV Damping Ring at the middle of injector linac aiming to lower positron emittance was deployed. The Phase-3 commissioning has started from spring 2019 and proceeds a maintenance during summer.

The electron/positron injector linac, with total length of 700 m, distributes beam to five rings (SuperKEKB HER/LER, two light source ring PF/PF-AR and one positron Damping Ring) with several parameters. The injector linac consists of 60 high-power accelerating units followed by a beam switch yard. The maximal beam repetition rate is 50 Hz and beam mode should switch every 20 ms [1].

TIMING SYSTEM ARCHITECTURE

The timing system in accelerator complex is to synchronize all relevant components and devices and precisely control some accelerator operations like injection and extraction, top-up and facility diagnostics with timestamp information. The precision of synchronization depends on the operation requirement as well as the size of the installation [2]. Less than 30 ps jitter is required in SuperKEKB and 300/700 ps in PF/PF-AR.

Event Timing

The idea of event timing control system is to transmit synchronized timing fiducial by encoding them as event codes in an event clock rate. Event clock is usually synchronized to a RF reference as well as phase locking to the mains voltage to keep beam intensity and quality.

The event timing system usually consists of event generator and event receiver. The event generator is able to accept software and hardware input and generate event signals and distribute with fanout to a number of event receivers by a multi-mode fibre network in star topology. The event receiver receives the event streams, recover the signals and generator event clocks that is phase locked to the event generator event clock.

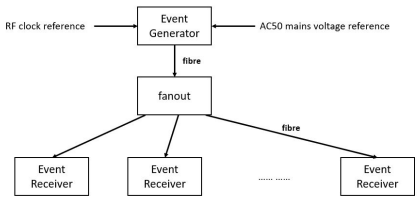

Figure 1: MRF event system.

We introduced event timing system produced by MRF company to meet our requirements. MRF timing system consists of "Event Generator (VME-EVG-230)" and "Event Receiver (VME-EVR-230-RF)". The event system uses optical fibre to transmit a 16-bit word at a frequency between 50 MHz and 142 MHz by 8b/10b encoding. In our case, event clock rate is 114.24 MHz in SuperKEKB linac, the bit rate of event link is about 2.3 GHz. Inside this 16-bit word, 8-bit distributed bus running in parallel and independent of the other 8-bit timing event allows distribution of eight signals updated with the event clock rate. All 256 event codes except for some special functions are user defined. Each EVR is able to generate pulse with an associated delay and width. MRF timing system is well integrated with EPICS control system by using mrfioc2 device support module [3]. Figure 1 is the scheme of MRF timing system.

* Work supported China Scholarship Council
† sdcswd@post.kek.jp

The selected event codes received in EVR will be decoded by 8b/10b and wrote into a 32-bit wide FIFO memory (8-bit event code and 24-bit timestamp) with attached timestamp information. This FIFO memory can hold up to 512 events. Write operation of an event to FIFO will trigger a VME interrupt. In mrfioc2 module, the EVR event is mapped to an EPICS event and make happen of several relevant EPICS records. Then, the IOC server sends the processed record information to all clients that subscribe to this record by Channel Access.

Simultaneous Injection

Based on EPICS and event-timing control, we use four virtual accelerators (VAs) for SuperKEKB project. A single injector linac would behave as four independent virtual accelerators with hundreds of independent parameters modulated pulse-by-pulse at 50 Hz (Fig. 2).

The equipment is operated via event-based, global, and synchronized controls to inject beams with different properties into four separate storage rings simultaneously. One of the crucial technologies used during the operation of linac is Pulse-to-Pulse Modulation (PPM) [4]. By means of PPM simultaneous top-up injection to several separated storage rings could be accomplished. Every pulse (20 ms) corresponds to a beam mode while every beam mode contains several event codes [5, 6] .

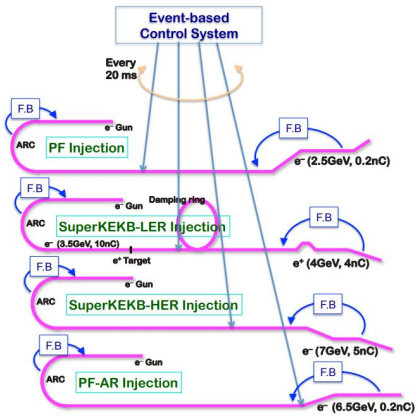

Figure 2: Single linac behaves as four virtual accelerators to inject their beams into four separate storage rings.

Human operator can set the beam mode sequence according to requirements of rings and EVG will generator event codes sequence as well as sending to EVR. Each event code has its own meaning, either for triggering particular component or indicating the pre-trigger signal of the next beam mode.

LOG SYSTEM

During the operation, linac must operate periodically in 50 Hz, thus many devices like power supply, pulsed magnets need to be triggered every 20 ms to assure bunches are accelerated at appropriate location. And only 2 ms width of fluctuation is accepted [7]. However, we noticed that sometimes klystron trigger time interval would be less than 18 ms and that would cause unstable beam and if beam bunch is not correctly transferred, in the worst situation, beam would hit the Belle II detector and damage it. Besides this, lack of a log system makes it hard to evaluate whether correct event codes and sequences are received by EVR or not. And during the 21st KEKB Accelerator Review Committee in 2016, it was suggested in the meeting to have an operational diagnostic program of event system. The collection of the event codes and timestamp information received in EVR is beneficial to diagnose abnormal events and avoid erroneously trigger. In order to improve the stability and maintenance of injector linac, an EPICS based log system is developed to record all the events information and compare them with beam mode pattern.

Data Acquisition

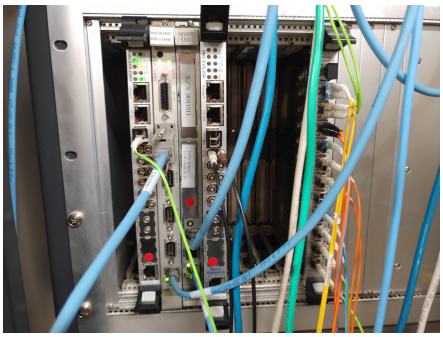

Figure 3: EVR card in VME crate.

The timing signal is generated and controlled at main timing station at injector linac and sent to EVRs all over SuperKEKB complex by fibre-optic cables. An EVR card used for log system is deployed in the VME crate placed in the control room of linac. Figure 3 is the picture of timing log system. MVME5500 on-board CPU controller and VxWorks-6.8.3 is used for developing EPICS IOC. A straightforward method to record all event codes used in injector linac is to create a number of EPICS PVs and monitor all of them. However, several mainly concerns are listed below:

- monitor a large number of PVs which updates very quickly might increase the network traffic in injector linac.
- camonitor call in Channel Access uses TCP to transmit the data packets. And owing to TCP's retransmission mechanism, events sequence recorded in client might not be consistent with sequence recorded in EVR.
- too many CA callback functions in IOC may increase the VME CPU load.

Consequently, we modified the device support of EPICS waveform record to store the event code and the EVR timestamp. Event code and timestamp pairs are stored in the waveform record and IOC server periodically scan this record and send it to client. The client program monitors the waveform

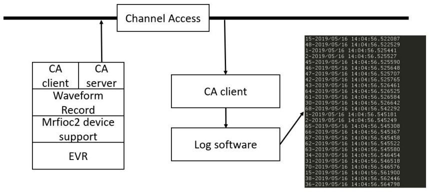

Figure 4: Event log system architecture.

record and saves data to hard disk. Figure 4 is the scheme of log system.

Log Performance

Some abnormal phenomenon are noticed during the log system test operation. First problem is the event software overflow of the EVR device support. According to the document of mrfioc2 EPICS support module, occurrences of certain "interested" event code number together with individual arrival time are placed in the EVR's FIFO buffer. A VxWorks task continuously fetch event one by one from the buffer and invoke a list of event-specified callback functions to notify some records to process. Since the invocation of callback function is processed in another thread, if the speed of event invocation is fast than callback function processing, then callback in other EPICS drivers may be lost. That means this event code can not be recorded in the waveform record [8].

Another problem is caused by processor issue. Some data would lose in the client during the periodically scanning of the waveform record. The CPU frequency of MVME5500 is 1 GHz which means 1 nanosecond per cycle, however, by contrast, the minimal interval between linac event code is 9 ns. According to the Fig. 5, sometimes the delay of scan-5 task is 4579 ms if we monitor all linac timing events. The scan-5 task scans a list of records every 5 second, in other words, if the delay of this task is large than 5 second, waveform record would not be scanned. From this perspective, we only registered some critical event codes to decrease the CPU load as a temporary solution.

NAME	ENTRY	TID	PRI	STATUS	PC	SP	ERRNO	DELAY
tJobTask	1ce564	4db1d0	0	PEND	24ec6c	4db0f0	0	0
tExcTask	1cd904	302970	0	PEND	24ec6c	302870	0	0
tLogTask	logTask	4de570	0	PEND	24ce28	4de430	0	0
tNbioLog	1cf238	4e1ea0	0	PEND	24ec6c	4e1d80	0	0
tShell0	shellTask	5f7d60	1	PEND	24ec6c	5f7a30	0	0
tShellRem6>	shellTask	614790	1	READY	257338	612990	ad0007	0
ipcom_tick>	261900	5bb410	20	DELAY	25615c	5bb390	0	91
tNet0	ipcomNetTask	4e6fc0	50	READY	24ec6c	4e6eb0	3d0001	0
ipcom_sysl>	17af48	5a2670	50	PEND	24f588	5a24c0	0	0
ipcom_teln>	ipcom_telnet	5dcb70	50	PEND	24ec6c	5dc970	0	0
ipsntps	1bab80	5dfd60	50	PEND+T	24ec6c	5dfbe0	3d0004	519514
ipcom_teln>	ipcom_telnet	1fffda80	50	PEND	24ec6c	1fffd860	0	0
tStdioProx>	18186c	1ffff4b0	50	READY	24ea64	1fffefb0	0	0
tLogin1fff>	181af0	6034c0	50	PEND	24f588	6033f0	0	0
tPortmapd	portmapd	5e3d10	54	PEND	24ec6c	5e3ac0	16	0
EVRFIFO	af4d00	634b60	109	READY	24ce28	634980	0	0
EVRFIFO	af4d00	c00ed0	109	READY	24ce28	c00cf0	0	0
cbHigh	af4d00	74c7b0	128	PEND	24ec6c	74c690	0	0
timerQueue	af4d00	6f73b0	129	PEND	24ec6c	6f7230	3d0004	0
scanOnce	af4d00	7df930	132	PEND	24ec6c	7df800	0	0
scan-0.1	af4d00	80f250	133	PEND+T	24ec6c	80f090	3d0004	28
scan-0.2	af4d00	808730	134	PEND+T	24ec6c	808570	3d0004	128
cbMedium	af4d00	743f90	135	PEND	24ec6c	743e70	0	0
scan-0.5	af4d00	801c10	135	PEND+T	24ec6c	801a50	3d0004	26
scan-1	af4d00	7fb0f0	136	PEND+T	24ec6c	7faf30	3d0004	80
scan-2	af4d00	7f45d0	137	PEND+T	24ec6c	7f4410	3d0004	1579
scan-5	af4d00	7edab0	138	PEND+T	24ec6c	7ed8f0	3d0004	4579
scan-10	af4d00	7e6f90	139	PEND+T	24ec6c	7e6dd0	3d0004	9579

Figure 5: Log system VxWorks IOC task information.

This Log System ran for a month and about 17 GB events data are collected. As Fig. 6 shows, however, event packets from CA server still lost about 1000 times in a roughly month. Such data lost issue is intolerable for the requirement of event timing Log System. So, we utilize several methods to upgrade the system.

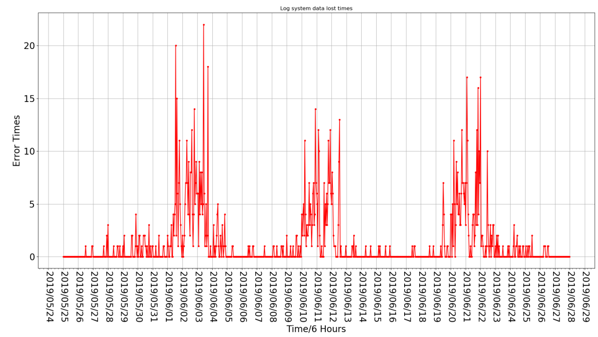

Figure 6: Event packet lost means that several PV values are not received by the client.

Log System Upgrading

To overcome the event overflow problem, we change and re-compile mrfioc2 module to satisfy our requirement by adding another high priority VxWorks task (parallel EPICS thread) to buffer the event code. When VME fetches an event code from EVR, this event code and timestamp will be put into a large-size ring-buffer and this ring-buffer put action wakes up a high priority thread. Resulting from this, all injector linac event codes can be recorded by the system without event software overflow error.

Several approaches have been adopted to solve the slow processor issue. A most straightforward method is to lower the CPU burden by reducing the number of EPICS records. Since EPICS performs initialization functions of many modules such as callbackInit, databaseInit and scanInit, and large amount of callback functions and records waiting for scanning would certainly increase the CPU load. Under these circumstances, only some EVR configuration records and event code I/O interrupt records are retained. Concerning this, a substitution data transmission approach, NFS protocol, is tested. Figure 7 is the scheme of Log System using NFS protocol.

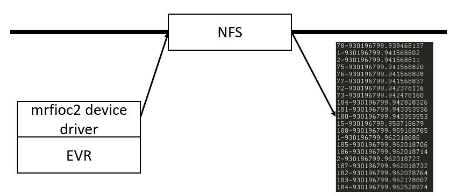

Figure 7: Event log system upgrading scheme. VxWorks is configured as an NFS client, a remote linac file system is mounted on it.

Network File System (NFS) is a distributed file system protocol, allowing a user on a client computer to access files over a computer network much like local storage is accessed [9]. The source code of mrfioc2 module is modified so that an EPICS thread is created aiming for directly write event code and its timestamp information to local hard disk. Under this circumstance, we utilize the low-level system call and communication protocol by using NFS comparing

to EPICS database record approach. Thus, the CPU load is dramatically decreased. Every day the log data will be processed and checked using Python script whether any abnormal beam sequences occur and the result of diagnosis will be sent by email to a list of email addresses automatically.

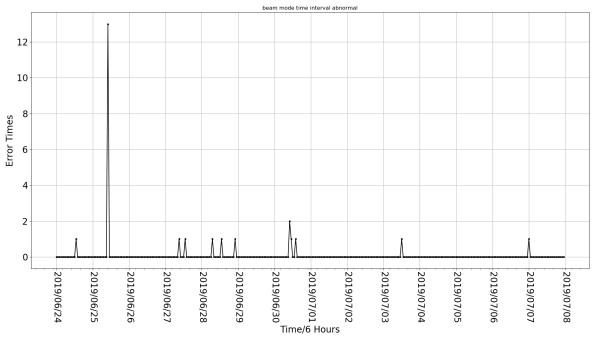

Figure 8: Abnormal beam mode interval information.

With the upgrading of old system, log system error is successfully eliminated. In this case, we can monitor every event along with individual timestamp sent from EVG located at main timing station. This log system can quickly detect and locate the fault, analyze the data and finally contribute to fix injection beam mode interval problems. Figure 8 is the picture of detected abnormal event sequences including continuous electron or positron beam mode and unusual beam mode time intervals. The reason of such anomaly event sequences is still under investigation.

SUMMARY AND OUTLOOK

In the near future, after the SuperKEKB injector linac operate on October, much more event code data will be saved and analyzed. Subsequently, we hope to utilize some the feature extraction algorithm to figure out the reason of abnormal beam mode intervals with the help of these event code timestamp information.

Besides, event-based timing system in injector linac is divided into functional two parts: upstream and downstream. Different event codes are distributed in two parts and received by two EVRs, upstream EVR and downstream EVR. Accordingly, we want to monitor both parts events in the VME and this would definitely increase VME's CPU load. Further experiments are undertaking now.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the financial support from China Scholarship Council.

REFERENCES

[1] K. Furukawa, M. Akemoto, D.A. Arakawa, Y. Arakida, H. Ego, A. Enomoto *et al.*, "Rejuvenation of 7-Gev SuperKEKB Injector Linac", in *Proc. IPAC'18*, Vancouver, BC, Canada, Apr. 4, pp. 300–303. doi:10.18429/JACoW-IPAC2018-MOPMF073

[2] T. Korhonen, "Review of Accelerator Timing Systems", in *Proc. 7th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'99)*, Trieste, Italy, Oct. 1999, paper FB1I01, pp. 167–170.

[3] MRF website, http://mrf.fi/index.php/timing-system

[4] K. Akai, K. Furukawa, H. Koiso, "SuperKEKB collider". https://doi.org/10.1016/j.nima.2018.08.017

[5] K. Furukawa, T. T. Nakamura, M. Satoh, and T. Suwada, "Pulse-to-pulse Beam Modulation and Event-based Beam Feedback Systems at KEKB Linac", in *Proc. 1st Int. Particle Accelerator Conf. (IPAC'10)*, Kyoto, Japan, May 2010, paper TUOCMH01, pp. 1271–1273.

[6] N. Higashi, S. Asaoka, K. Furukawa, K. Haga, K. Harada, T. Higo *et al.*, "Construction and Commissioning of Direct Beam Transport Line for PF-AR", in *Proc. 8th Int. Particle Accelerator Conf. (IPAC'17)*, Copenhagen, Denmark, May 2017, paper WEPAB044, pp. 2678–2680.

[7] H. Kaji, K. Furukawa, M. Iwasaki, and E. Kikutani *et al.*, "Bucket selection system for SuperKEKB", in *Proceedings of 12th Annual Meeting of PASJ*, Fukui, Japan, 2015, paper THP100.

[8] Mrfioc2 website, http://epics.sourceforge.net/mrfioc2

[9] *vxworks kernel programmers guide 6.8*, Wind River, Alameda, CA, USA, Nov. 2009, pp. 433-440.