ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA105

BEAM SYNCHRONOUS DATA ACQUISITION USING THE VIRTUAL EVENT RECEIVER

G. Mun, W. W. Lee, G. Kim, J. Hu, C. B. Kim, H. S. Kang
Pohang Accelerator Laboratory, Pohang, Korea

Abstract

The 4th generation light source, PAL-XFEL, is an X-ray free electron laser in Pohang, Korea. One of key features of the event timing system in the PAL-XFEL, the beam synchronous acquisition is used in many beam diagnostics and analysis and the species of that increase gradually. In order to reduce the cost for event receivers which are required for operating the beam synchronous acquisition and to resolve the difficulty of the limited platform dependant on event receivers, we developed the virtual event receiver system receiving timestamps and BSA information from an event generator not using real event receivers. In this paper, we introduce the software architecture of the virtual event receiving system and present test results of it.

BEAM SYNCHRONOUS ACQUISITION

Event timing system is one of the key infra systems to control accelerators, which basically controls triggers for each instrument and system in an accelerator system. The event timing system generally consists of event generators and event receivers. The event generator, EVG, generates event definitions which are scheduled in a time sequence according to the operation policy in an accelerator system. The event receiver, EVR, generates trigger signals for clients such as modulators, instrument for diagnostics, and low level RF controllers based on the information transferred from the EVG. This event information includes each timestamp corresponding to each event. The event timing system of the PAL-XFEL consists of the hardware implemented by the MicroResearch Finland Co. and the software developed for the LCLS [1-3].

The event timing system of the PAL-XFEL supports the beam synchronous acquisition, BSA, which is useful for analysing the correlations among several client devices by collecting data in terms of same beam pulse as realized in the LCLS [4]. Figure 1 shows a schematic drawing to describe the concept of BSA by using the EVG and EVRs. Each client input output controller, IOC, has an EVR which receives event pattern information from the EVG IOC.

Figure 2 shows the orbit correction system used in PAL-XFEL. This orbit correction system uses the real-time beam position data taken in BSA by the beam position monitors. The BSA is also used to measure beam profiles by using the wire scanners in the PAL-XFEL, which is especially essential to correct the jitter of the raw data from the wire scanners.

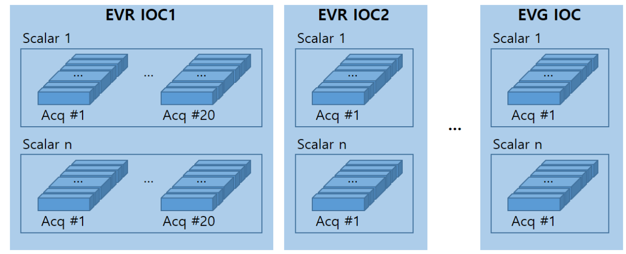


Figure 1: Data acquisition among IOCs in an event timing system using EVRs and an EVG.

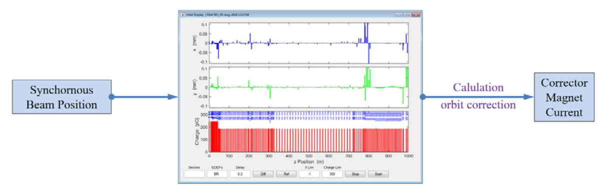


Figure 2: Orbit correction system using BSA to take synchronous beam positons with the beam position monitors in the PAL-XFEL.

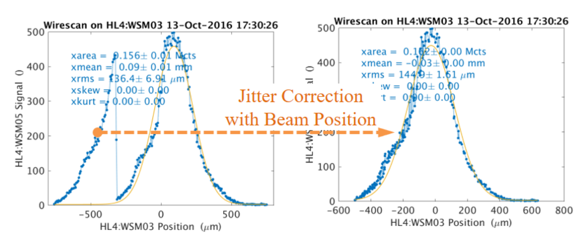


Figure 3: Typical example of beam profile measurement with wire scanner using BSA.

VIRTUAL EVENT RECEIVER

The EVR has three major roles in the event system of the PAL-XFEL. The first one is to generate trigger signals. The generation of trigger signal are processed by the FPGAs embedded in the EVR. The other functions are to transfer the information related with the timestamp and BSA from the EVG as mentioned before.

The actual works to generate timestamps and taking data in BSA can be processed by the CPU of an IOC. The EVRs are simply interconnect the information related with the events between the EVG and the client IOCs or the processors. Therefore, if we realize the function of delivering information related with BSA and timestamp by using a separated simple micro-computer and if the client system can make the trigger signal based on the received event data

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA105

from the EVG, we can make the client system work as if an EVR is running in the client system.

Figure 3 shows the proposed scheme, namely virtual EVR. The virtual EVR concept requires a software to process transferring the EVG information instead of a real EVR hardware. In an actual EVR, the EVR's controlling CPU transfers the received EVG's buffer data through an interrupt service routine, ISR. In the virtual EVR, this is processed by the publisher's CPU and the subscriber's CPU through another communication network. We also make the EVG do not any tasks related with the virtual EVR to guarantee the reliability of the event timing system.

The virtual EVR publisher sends the buffer data received from the EVG to the virtual EVR subscribers within 10 µs. The virtual EVR subscriber invokes an ISR using the data stream to make timestamps and to make clients run in BSA mode. Since a publisher may has n number of subscribers as shown in Fig. 4, it is possible to construct cost effective BSA system with the virtual EVR system. We realized this virtual EVR's publisher and subscriber software based on the zeroMQ which supports most platform in free [5].

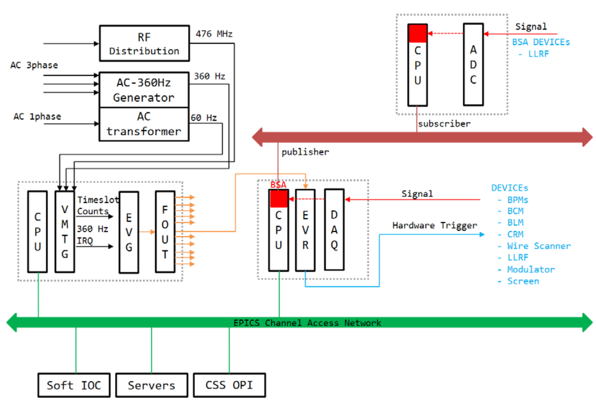


Figure 4: Concept of virtual event receiver.

Development & Experiment

The existing EVR driver software for a real EVR device provides a powerful diagnosis tool for the event timing system, which is also useful for debugging the virtual EVR system. Figure 5 shows a typical screen image of the EVR diagnostics software at the case of the virtual EVR. We verified the performance of the virtual EVR subscribers by checking the fiducial delay corresponding to the receiving time of the event code, 1, transferred with the rate of 360 Hz. While the deviation of the average of that is very small, the difference between the maximum and the minimum of the fiducial delay is large, which is due to the reason that the virtual EVR was developed and tested on a virtual Linux PC environment which may occur unexpected latency.

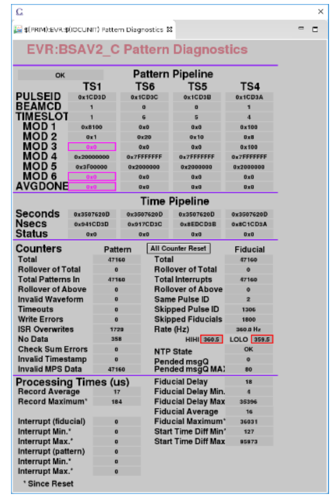


Figure 5: Screen image of virtual EVR subscriber pattern diagnostics. Fiducial Delay = 18 µs, Fiducial Delay Min. = 4 µs, Fiducial Delay Max. = 35396 µs, Fiducial Avg. = 16 µs.

Figure 6 shows the test result of the virtual EVR which uses the virtual EVR subscriber implemented on a Raspberry Pi which is a cheap but compact enough to test it.

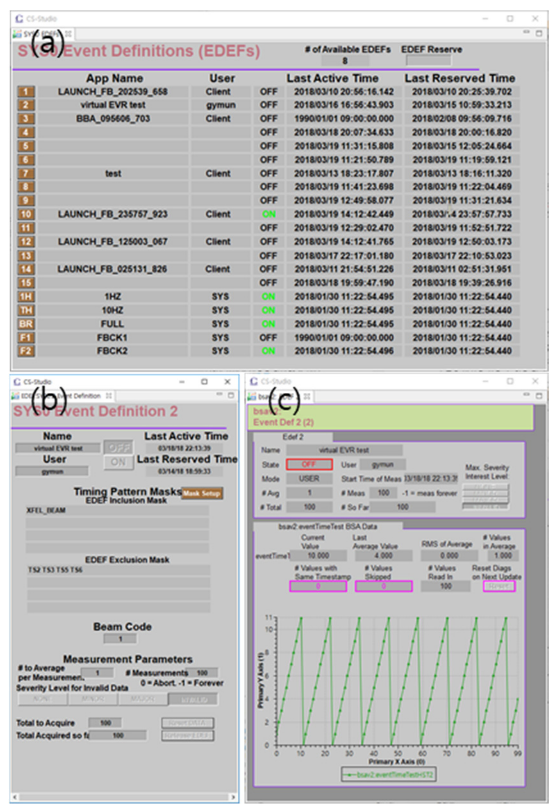


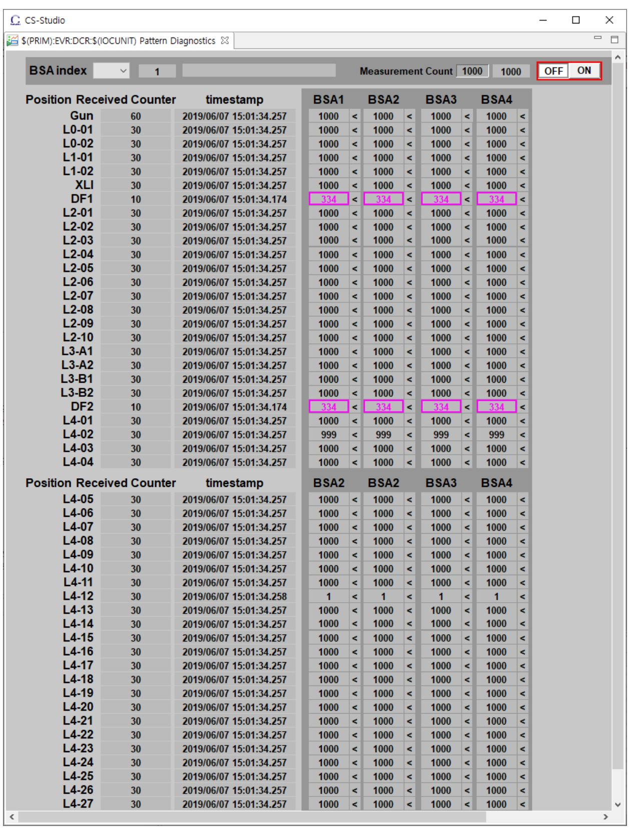
Figure 6: Screen images of the BSA test by using a virtual EVR on a Raspberry Pi. (a) Defining an event for a BSA. (b) Setting detail parameters for the BSA. (c) Monitoring the test result.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA105

Virtual EVR for LLRF Controller

52 RF klystron modules are installed in the linear accelerator the PAL-XFEL. Since each RF module is controlled and monitored by a low level RF controller, namely LLRF, 52 LLRFs are working in the PAL-XFEL [6]. The LLRF of the PAL-XFEL does not have an EVR device unlike other BSA devices such as beam position monitors, but was designed to work as a virtual EVR subscriber so as to evaluate BSA function in the LLRF.

Figure 7 shows the screen image of the diagnostics of BSA for the all LLRFs. This software was developed by using the GUI of the Control System Studio, CSS [7]. The received count number means the number of data streams per second, which shows that the each RF module runs in proper pulse rate. The timestamps are based on the event buffer stream transferred from the EVG, representing a global time base. Because we set the number of data to store in a BSA buffer is 1000 even though the maximum value is 2800, the numbers of the four BSA data, BSA1, BSA2, BSA3, and BSA4 are all 1000 in this case and the every LLRF gathers 1000 data for each BSA.



Position	Received Counter	timestamp	BSA1		BSA2		BSA3		BSA4	
Gun	60	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L0-01	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L0-02	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L1-01	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L1-02	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
XLI	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
DF1	10	2019/06/07 15:01:34.174	334	<	334	<	334	<	334	<
L2-01	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L2-02	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L2-03	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L2-04	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L2-05	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L2-06	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L2-07	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L2-08	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L2-09	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L2-10	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L3-A1	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L3-A2	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L3-B1	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L3-B2	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
DF2	10	2019/06/07 15:01:34.174	334	<	334	<	334	<	334	<
L4-01	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-02	30	2019/06/07 15:01:34.257	999	<	999	<	999	<	999	<
L4-03	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-04	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<

Position	Received Counter	timestamp	BSA2		BSA2		BSA3		BSA4	
L4-05	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-06	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-07	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-08	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-09	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-10	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-11	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-12	30	2019/06/07 15:01:34.258	1	<	1	<	1	<	1	<
L4-13	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-14	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-15	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-16	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-17	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-18	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-19	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-20	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-21	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-22	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-23	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-24	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-25	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-26	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<
L4-27	30	2019/06/07 15:01:34.257	1000	<	1000	<	1000	<	1000	<

Figure 7: Screen image of the diagnostics of BSA for the all LLRFs.

Figure 8 shows the test example of beam energy jitter by using the BSA data from the LLRFs and the beam position monitors in the PAL-XFEL. The amplitude and the phase were acquired by the LLRFs with the virtual EVR, but the dump beam positions were acquired by the beam position monitors with hardware type EVR.

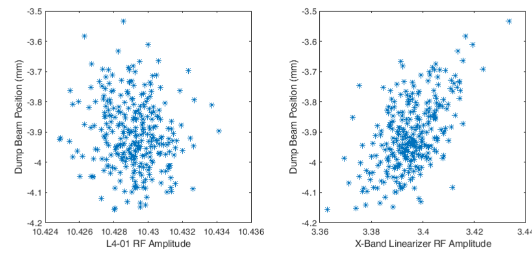


Figure 8: Correlation check RF vs Beam Energy (Dump BPM).

CONCLUSION

We developed a virtual EVR system which can make the system without physical EVR hardware by using a software consisted of the virtual EVR publisher and the virtual EVR subscribers requiring an independent network communicating the event information related with the time stamp and the BSA. Because it is independency from platform, this architecture has advantages in cost saving and extendibility, comparing to the hardware type EVR, even though it needs independent network resource.

REFERENCES

[1] Micro-Research Finland, `http://www.mrf.fi`

[2] J. E. Dusatko, S. Allison, J. Browne, and P. Krejcik, "The LCLS Timing Event System", in *Proc. 14th Beam Instrumentation Workshop (BIW'10)*, Santa Fe, NM, USA, May 2010, paper TUPSM083, pp. 379-383.

[3] P. Krejcik *et al.*, "Timing and Synchronization at the LCLS", in *Proc. 8th European Workshop on Beam Diagnostics and Instrumentation for Particle Accelerators (DIPAC'07)*, Venice, Italy, May 2007, paper WEPC28, pp. 373-375.

[4] K. H. Kim, S. Allison, T. Straumann, and E. Williams, "Real-Time Performance Improvements and Consideration of Parallel Processing for Beam Synchronous Acquisition (BSA)", in *Proc. 15th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'15)*, Melbourne, Australia, Oct. 2015, pp. 992-994, `doi:10.18429/JACoW-ICALEPCS2015-WEPGF122`

[5] ZeroMQ, `http://zeromq.org`

[6] J. Hu *et al.*, "LLRF Development for PAL-XFEL", in *Proc. 7th Int. Particle Accelerator Conf. (IPAC'16)*, Busan, Korea, May 2016, pp. 2761-2764. `doi:10.18429/JACoW-IPAC2016-WEPOR040`

[7] Control System Studio, `http://controlsystemstudio.org`