





Authors: Longwei Lai, <u>Yongbin Leng*</u>, Yingbing Yan, Weimin Zhou, Fangzhou Chen#, Ning Zhang SSRF, Shanghai, P.R. China

Abstract

The development of Digital BPM Signal Processors (DBPM) for SSRF started from 2008. The first prototype for SSRF storage ring was completed in 2012, with turn-by-turn resolution better than 1µm. From 2016 to 2017, SSRF successively constructed two FEL facilities in China, DCLS and SXFEL test facilities. The second version DBPM was developed and used in large scale during this period to meet the requirements of signal processing for stripline BPMs and cavity BPMs. After that, we turned to the development of DBPM for SSRF storage ring based on the second version hardware, including FPGA firmware, EPICS IOC, EDM control panel. The development was completed and tests were carried out in early 2018. Test results showed that the position data is accurate and can monitor beam movement correctly, and online turn-by-turn position data resolution reaches 0.46µm. This paper will introduce the design of DBPM for the SSRF storage ring and the tests carried out to verify the data accuracy and evaluate the system performance.

•Firmware and software design



•Verification tests on SSRF SR



Counts

-50 0 SA Y Position read

SA 10Hz resolution 43nm

Performance tests





45 44 43 02 0.1 0.1 02 0.3 0.4 0.5 FA 10kHz resolution 0.10μm

Conclusion

Online tests were carried out in early 2018 on SSRF. Test results show that DBPM processor can measure the beam orbit accurately when comparing the results with Libera Brilliance. The turn-by-turn data resolution can reach 0.46 µm rms, much better than the first version.

We can draw the conclusion from above results that the new version DBPM meets the requirement of beam orbit measurements on the storage ring, and can be used for beam position control. However, more time and resources are still needed to make the DBPM more stable before final deployment.

Gratefully acknowledge the paper guidance from Jeff Corbett at SLAC.

IBIC'18, Shanghai, China

